User interface

- Usability
  - Easy to understand, easy to handle
- User-friendliness
  - Equipment having highly-usable interface
- Human-centered design

Paradigms for interaction (1)

1. Time-sharing
2. Video display units
3. Programming toolkits for complex interactive systems
4. Personal computing
5. Window systems and the WIMP (window, icons, menus and pointers) interface
6. The metaphor
7. Direct manipulation, WYSIWYG paradigm

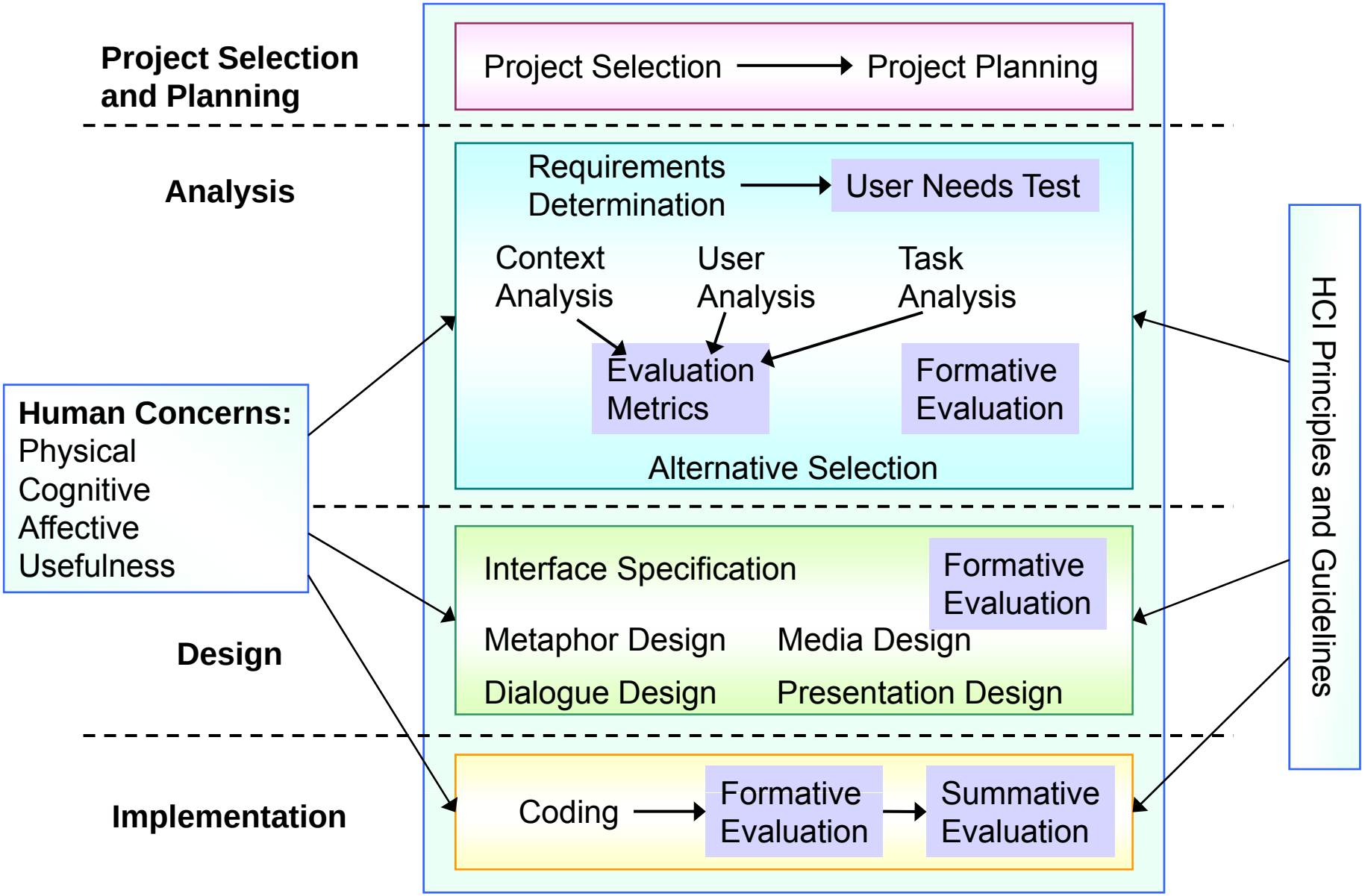
Features of a direct manipulation interface
(Ben Shneiderman)

1. Visibility of the objects of interest
2. Incremental action at the interface with rapid feedback on all actions
3. Reversibility of all actions, so that users are encouraged to explore without severe penalties
4. Syntactic correctness of all actions, so that every user's action is a legal operation
5. Replacement of complex command languages with actions to manipulate directly the visible objects (and, hence, the name direct manipulation)

Paradigms for interaction (2)

8. Language versus action (generic and repeatable procedures, e.g. information retrieval systems)
9. Hypertext and hypermedia
10. Multi-modality
11. Computer-supported cooperative work (CSCW, groupware)
12. The World Wide Web (WWW)
13. Agent-based interfaces
14. Ubiquitous computing

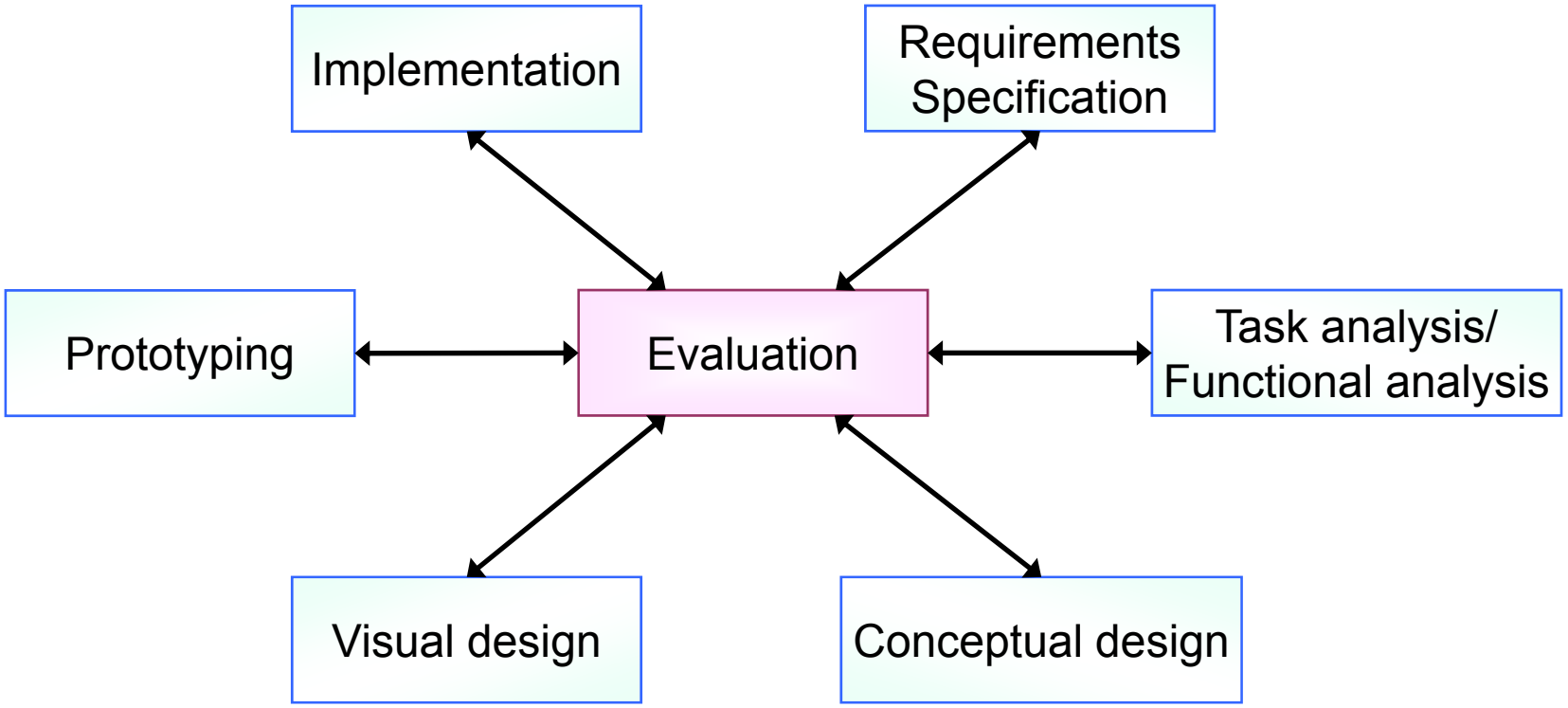
The HCI development methodology.



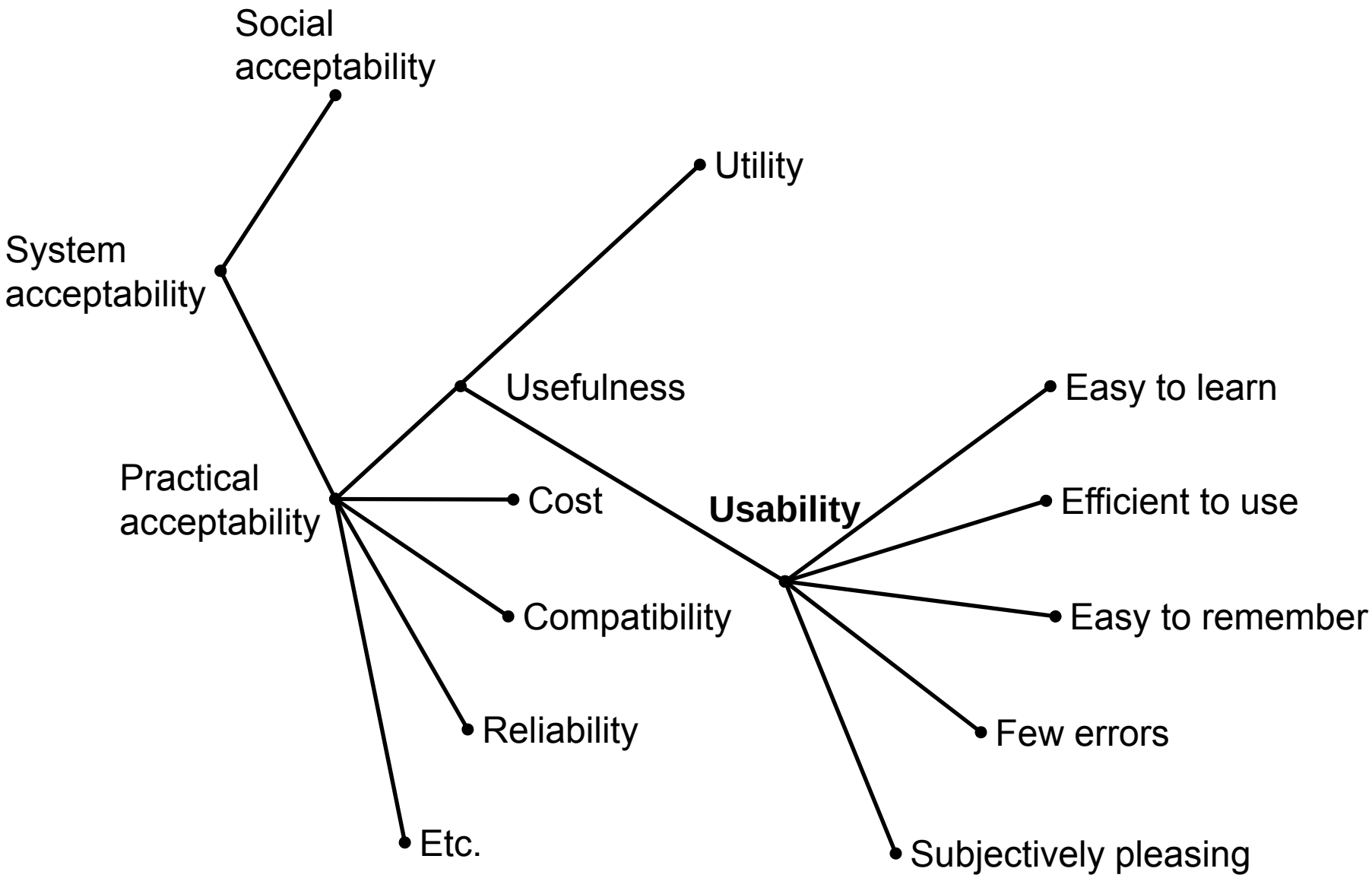
Multiple Concerns of HCI

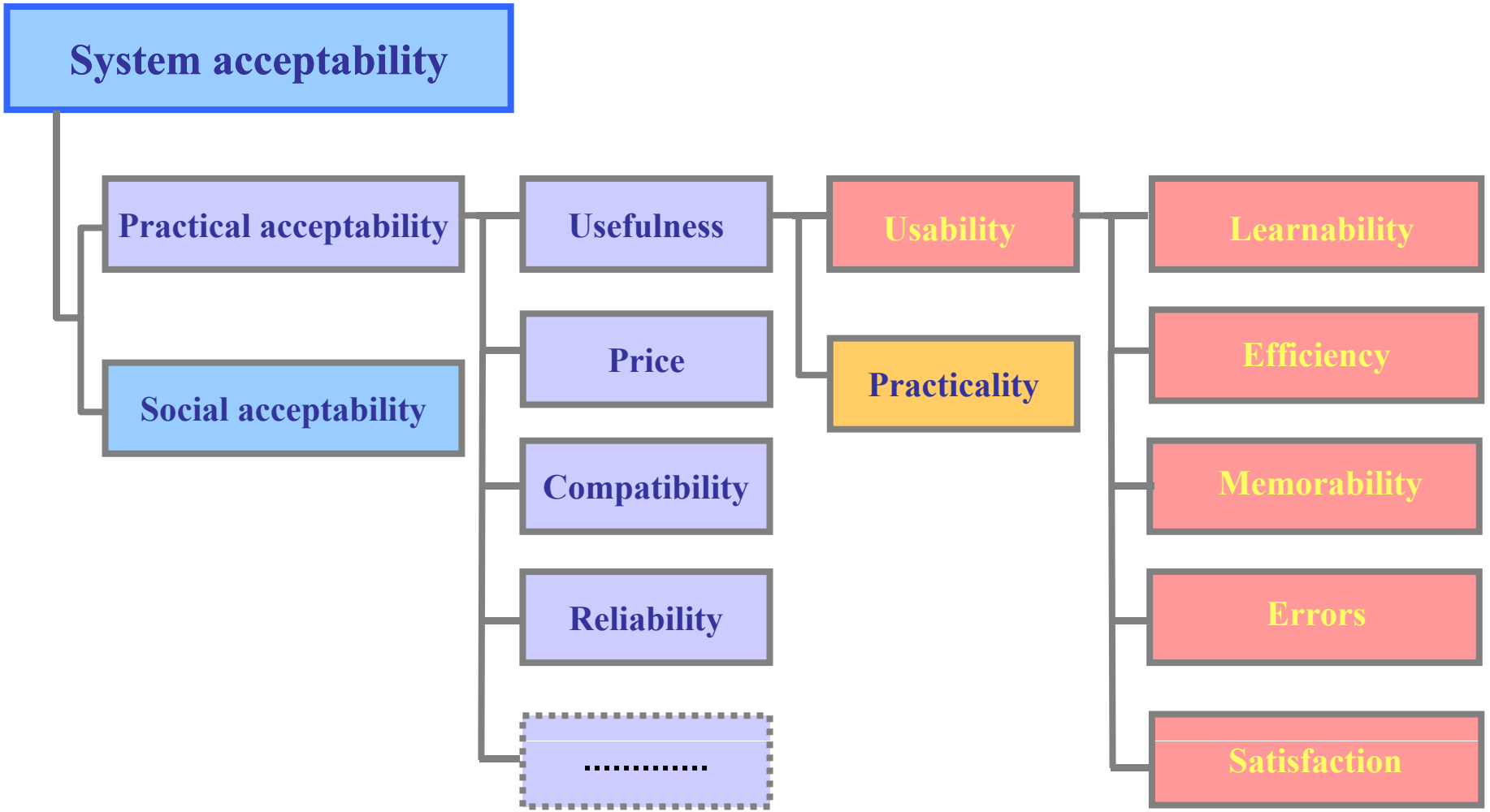
HCI Concern	Description	Sample Measure Items
Physical	System fits our physical strengths and limitations and does not cause harm to our health	Legible Audible Safe to use
Cognitive	System fits our cognitive strength and limitations and functions as the cognitive extension of our brain	Fewer errors and easy recovery Easy to use Easy to remember how to use Easy to learn
Affective	System satisfies our aesthetic and affective needs and is attractive for its own sake	Aesthetically pleasing Engaging Trustworthy Satisfying Enjoyable Entertaining Fun
Usefulness	Using the system would provide rewarding consequences	Support individual's tasks Can do some tasks that would not be possible without the system Extend one's capability Rewarding

Evaluation as the center of systems development

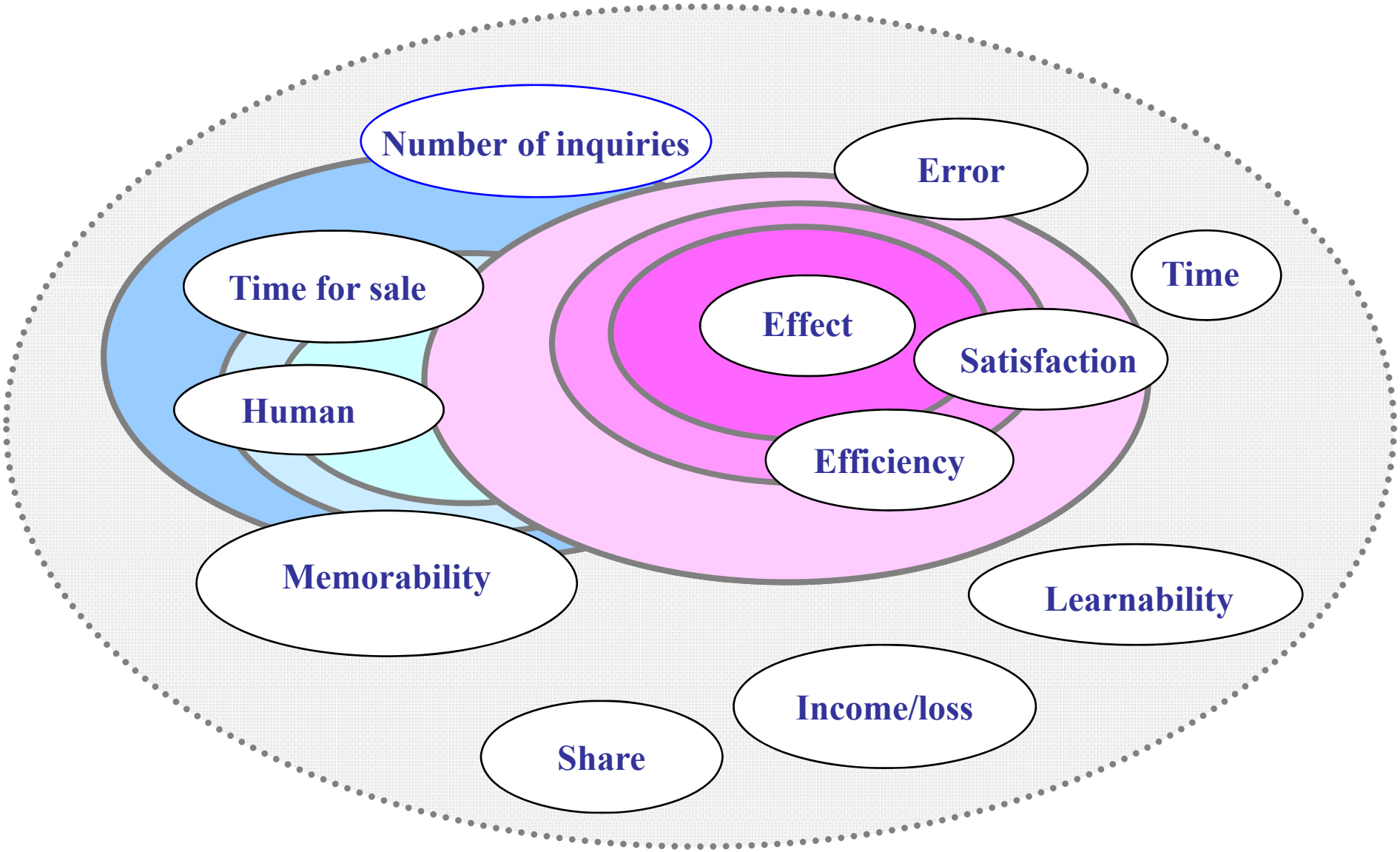


System acceptability and usability (Nielsen)



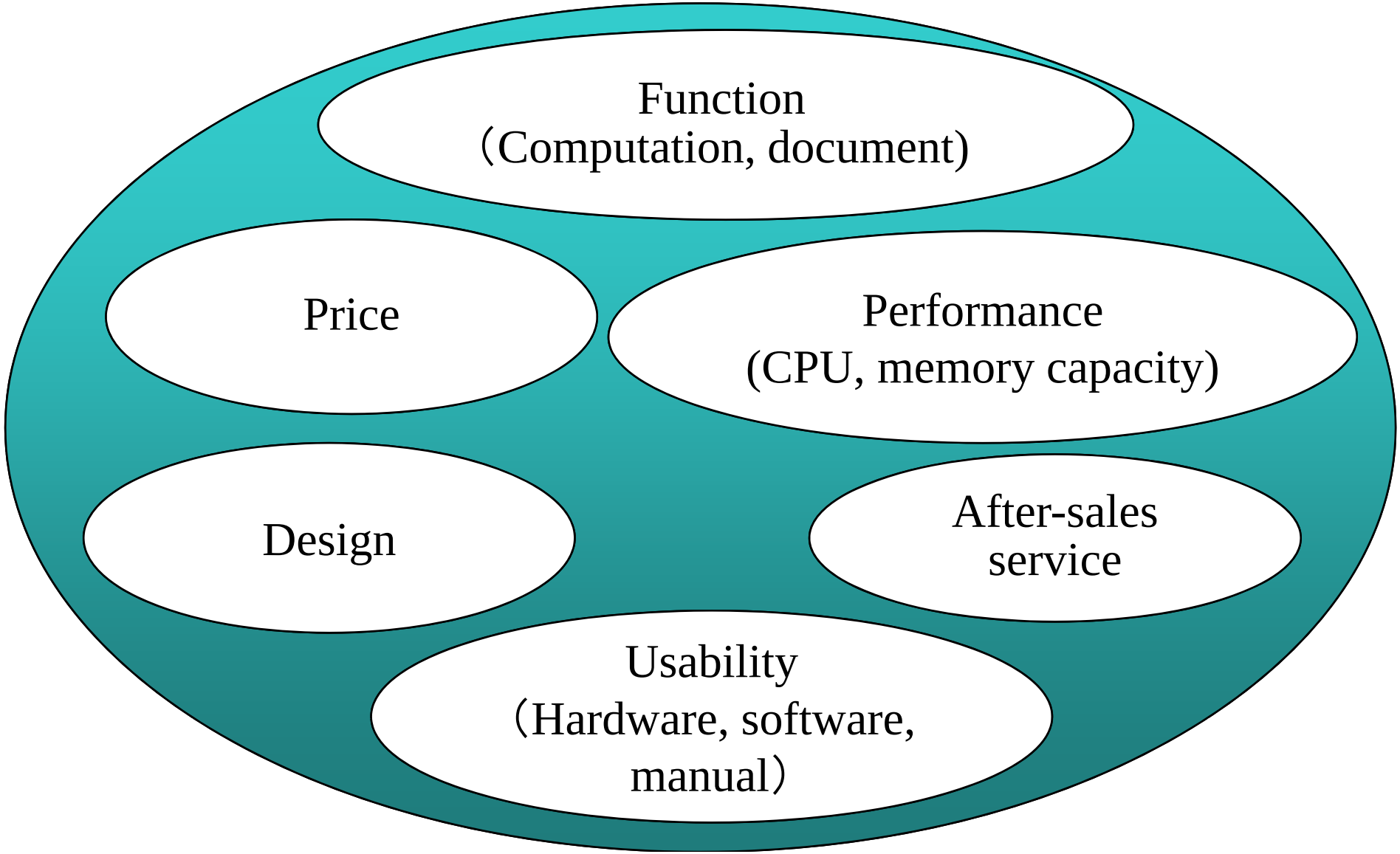


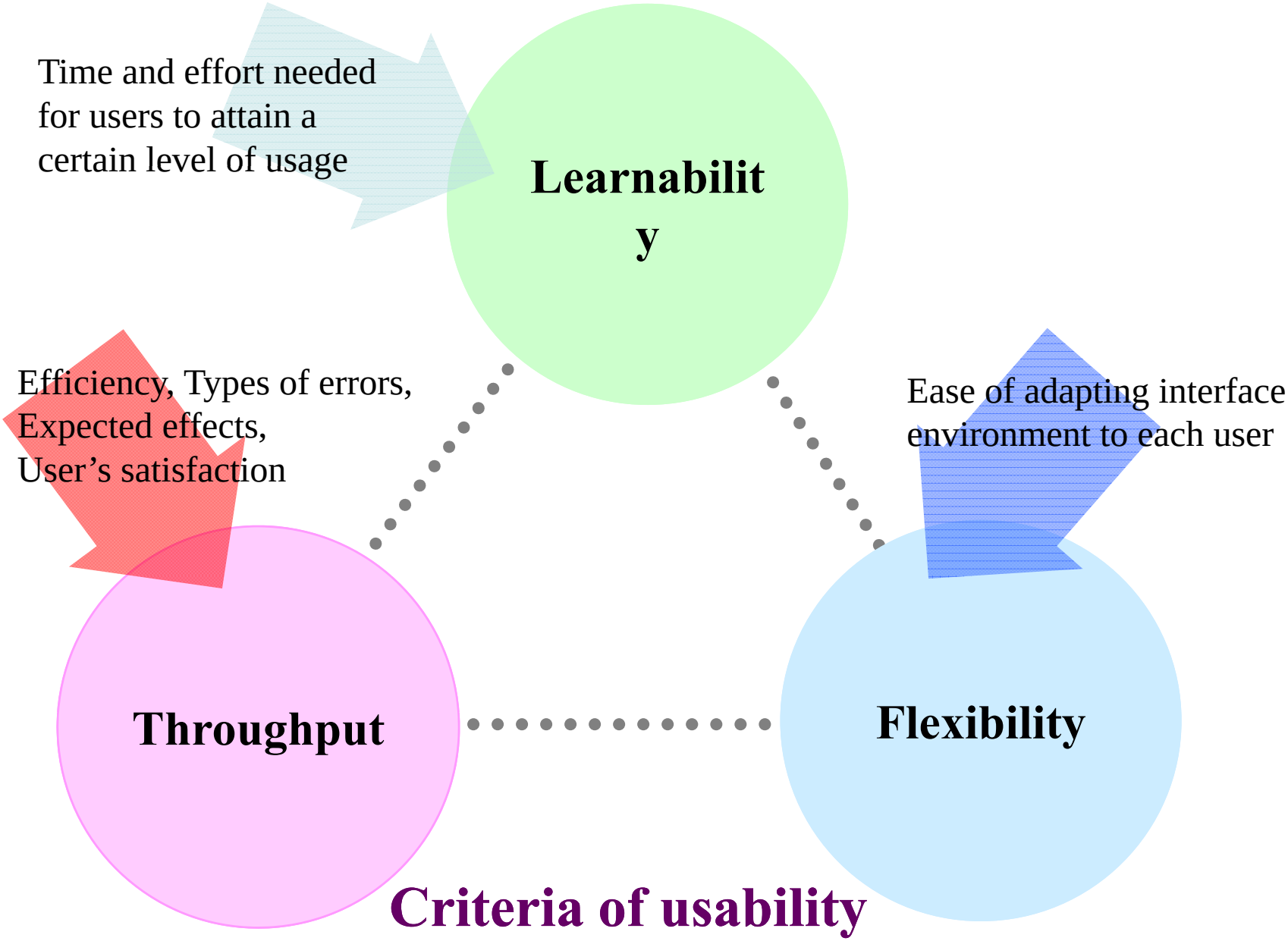
Attribute models and usability for systems to be accepted (Nielsen)



Factors related to usability

Elements which compose the value of electrical appliances (In the case of PC)





Criteria of usability

Principles to support usability

1. **Learnability:** the ease with which new users can begin effective interaction and achieve maximal performance

2. **Flexibility:** the multiplicity of ways the user and system exchange information

3. **Robustness:** the level of support provided to the user in determining successful achievement and assessment of goals

Summary of principles affecting learnability

Principle	Definition	Related principles
Predictability	Support for the user to determine the effect of future action based on past interaction history	Operation visibility
Synthesizability	Support for the user to assess the effect of past operations on the current state	Immediate/eventual honesty
Familiarity	The extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system	Guessability, affordance
Generalizability	Support for the user to extend knowledge of specific interaction within and across applications to other similar situations	
Consistency	Likeness in input-output behavior arising from similar situations or similar task objectives	

Summary of principles affecting flexibility		
Principle	Definition	Related principles
Dialog initiative	Allowing the user freedom from artificial constraints on the input dialog imposed by the system	System/user pre-emptiveness
Multi-threading	Ability of the system to support user interaction pertaining to more than one task at a time	Concurrent vs. interleaving, multi-modality
Task migratability	The ability to pass control for the execution of a given task so that it becomes either internalized by user or system or shared between them	
Substitutivity	Allowing equivalent values of input and output to be arbitrarily substituted for each other	Representation multiplicity, equal opportunity
Customizability	Modifiability of the user interface by the user or the system	Adaptivity, adaptability

Summary of principles affecting robustness		
Principle	Definition	Related principles
Observability	Ability of the user to evaluate the internal state of the system from its perceivable representation	Browsability, static/dynamic defaults, reachability, persistence, operation visibility
Recoverability	Ability of the user to take corrective action once an error has been recognized	Reachability, forward/backward recovery, commensurate effort
Responsiveness	How the user perceives the rate of communication with the system	Stability
Task conformance	The degree to which the system services support all of the tasks the user wishes to perform and in the way that the user understands them	Task completeness, task adequacy

8 Golden Rules (Shneiderman)

Principle	Content	Example
(1) Consistency	Consistent control for similar setting. Keep exceptions to be minimum.	Use same terms in prompt messages, menu, and help document. Use same command format for the whole system.
(2) Short-cut for a heavy user	Provide mechanisms to minimize the number of dialogues and inputs for heavy users. Let user decide using short-cut or not.	Abbreviation : input command 'print' → 'pr' and 'p' Special key: assign commands for function keys Hidden command : functions that are convenient but inappropriate from a consistency viewpoint are hidden for general users Macro function : unite multiple commands
(3) Useful feedback	Provide feedback to every action.	Simple responses are provided to frequently used and less important actions, and large amount of information is provided to important and less frequently used actions.

Principles	Contents	Examples
(4) Phased sense of achievement	Flow of operation should be well-organized. Give sense of satisfaction and ease step by step.	When the task is complicated and the user has a possibility to get lost, the task flow should be displayed on the screen and present where the user is.
(5) Easy error processing	Prevent user's fatal error. Even if it happens, an easy correction procedure is provided.	Re-input of command to correct errors need to be limited only to the error parts. Describe how to cope with the error in combination with the error message
(6) Reversible operation	Operations should be reversible, so that a user can trace back the system states.	Provide functions for the user to go back to the previous state, such as 'undo', when an error has occurred.
(7) User-centered control	Let the user not passively respond, but have initiative of operations. Give user a feeling that he/she controls the system.	Should not give such a response that could surprise the user. Should not require the user a large amount of data input. Provide only necessary information to the user by a simple operation. Provide operations which can be expected by the user.
(8) Light short-term memory load	Minimize the amount of information that the user needs to memorize for operations.	Display related information in a single page. Reduce the operations which are irrelevant to primary tasks, such as moving and changing the size of windows. Display information needed for the task on the screen, or make the user possible to look at them by simple operations.

Eight Golden Rules for User Interface Design (Shneiderman) -1

Rules	Description
Strive for consistency	This rule is the most frequently violated one, but following it can be tricky because there are many forms of consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; consistent color, layout, capitalization, fonts, and so on should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number.
Cater to universal usability	Recognize the needs of diverse users and design for plasticity, facilitating transformation of content. Novice-expert differences, age ranges, disabilities, and technology diversity each enrich the spectrum of requirements that guides design. Adding features for novices such as explanations, and adding features for experts, such as shortcuts and faster pacing, can enrich the interface design and improve perceived system quality
Offer informative feedback	For every user action, there should be some system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly.

Eight Golden Rules for User Interface Design (Shneiderman)-2

Rules	Description
Design dialogs to yield closure	Sequence of actions should be organized into groups with a beginning, middle, and end. Informative feedback at the completion of a group of actions gives operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans from their minds, and a signal to prepare for the next group of actions.
Prevent errors	As much as possible, design the system so that users cannot make serious errors. If a user makes an error, the interface should detect the error and offer simple, constructive, and specific instructions for recovery. Erroneous actions should leave the system state unchanged, or the interface should give instructions about restoring the state.
Permit easy reversal of actions	As much as possible, actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone, thus encouraging exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name and the address book.

Eight Golden Rules for User Interface Design (Shneiderman)-3

Rules	Description
Support internal locus of control	Experienced operators strongly desire the sense that they are in change of the interface and that the interface responds their actions. Surprising interface actions, tedious sequences of data entries, inability to obtain or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction.
Reduce short-term memory load	The limitation of human information processing in short-term memory requires that displays be kept simple, multiple-page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequence of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

Usability heuristics

Items	Details
Simple and natural dialogue	Dialogue should not include unnecessary information. All the information should be presented in a natural and logical order.
Use user's own words	Terms and concepts used in the dialogue should be familiar with the users.
Keep user's memory load minimum	Information necessary for actions should be always presented or easily accessible.
Keep consistency	Different terms, states or actions should not be used for the same matter.
Offer feedback	Offer appropriate feedback at appropriate timing to let the user know the present situation of the progress of actions.
Clearly show exits	Prepare exits in a standard dialogue to escape from inappropriate states.
Offer short-cut	Satisfy both beginners and experts by providing accelerator.
Good error messages	Need to use simple language, accurately describe problems, and suggest solutions.
Prevent errors	Prevent user's error in advance.
Help and document	Need to be easy to search, and briefly describe the procedures from the viewpoint of users' actions.

[Nielsen 1990]

Ten Usability Heuristics-1 (Nielsen)

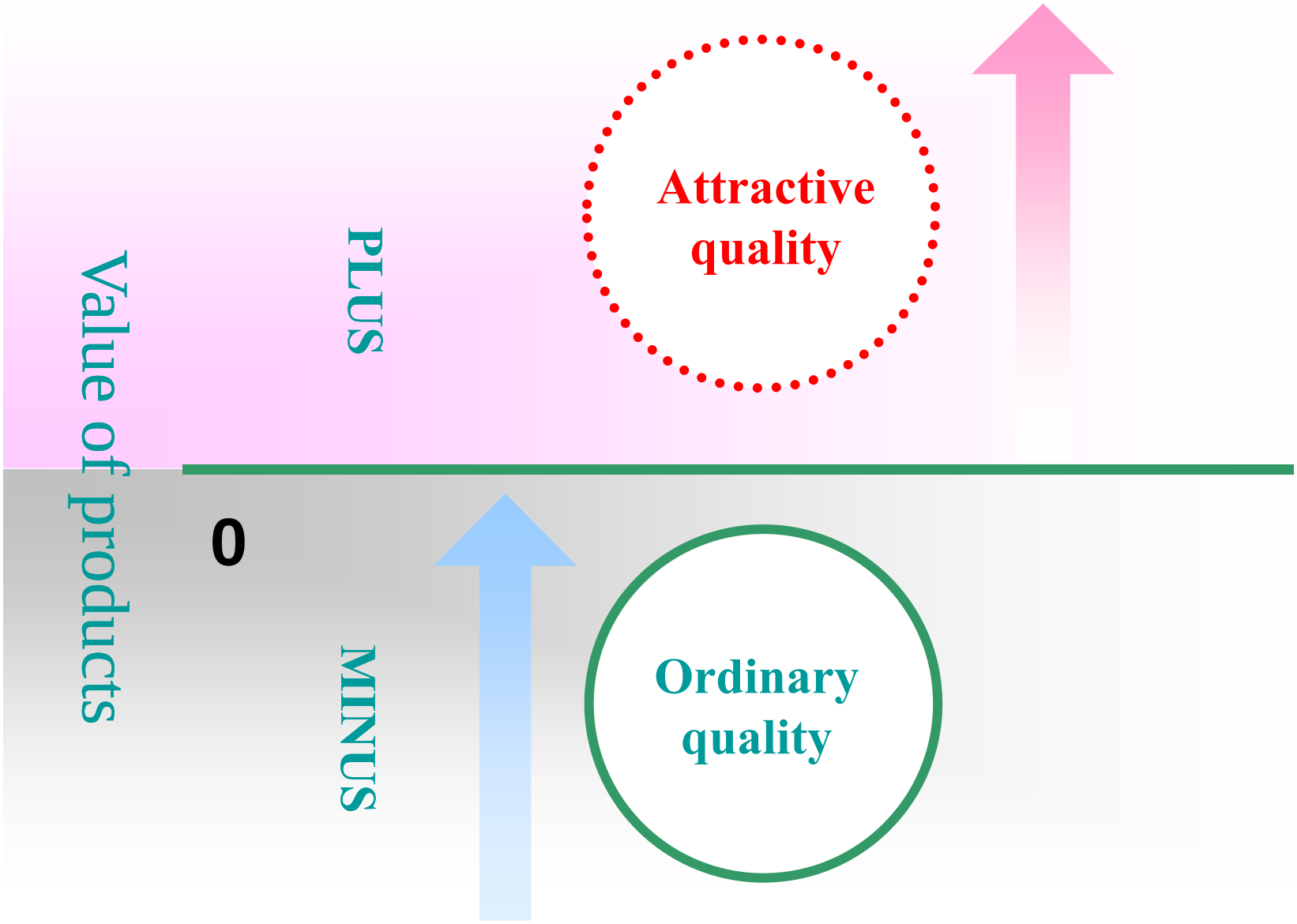
Rules	Description
Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Match between system and the real world	The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
Error prevention	Even better than good messages is a careful design that prevents a problem from occurring in the first place.

Ten Usability Heuristics-2 (Nielsen)

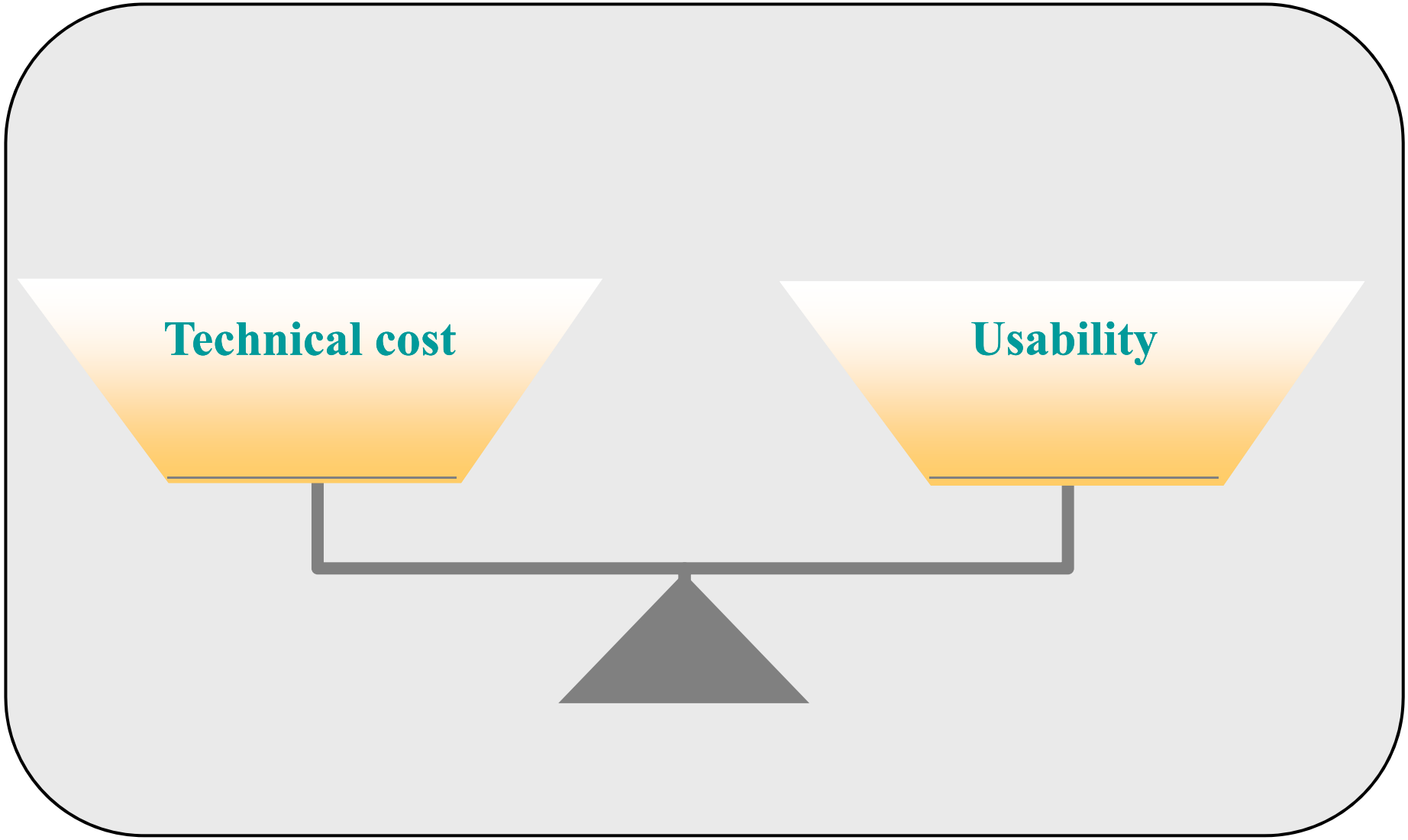
Rules	Description
Recognition rather than recall	Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
Flexibility and efficiency of use	Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalist design	Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, be focused on the user's task, list concrete steps to be carried out, and not be too large.

HOMERUN Heuristics for Commercial Web sites (Nielsen)

Description
• **H**igh-quality content
• **O**ften update
• **M**inimal download time
• **E**ase of use
• **R**elevant to users' medium
• **U**nique to the online medium
• **N**et-centric corporate culture

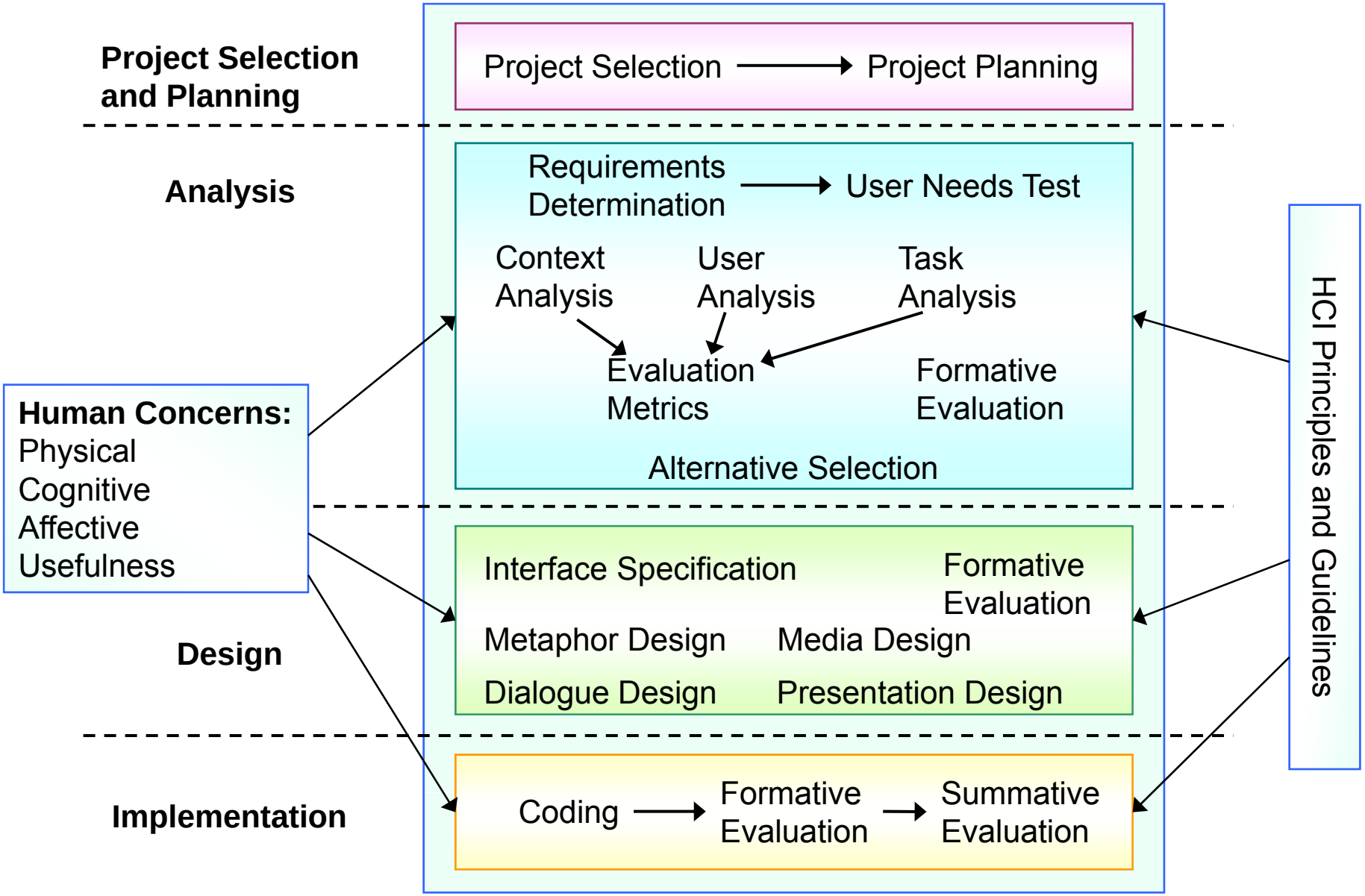


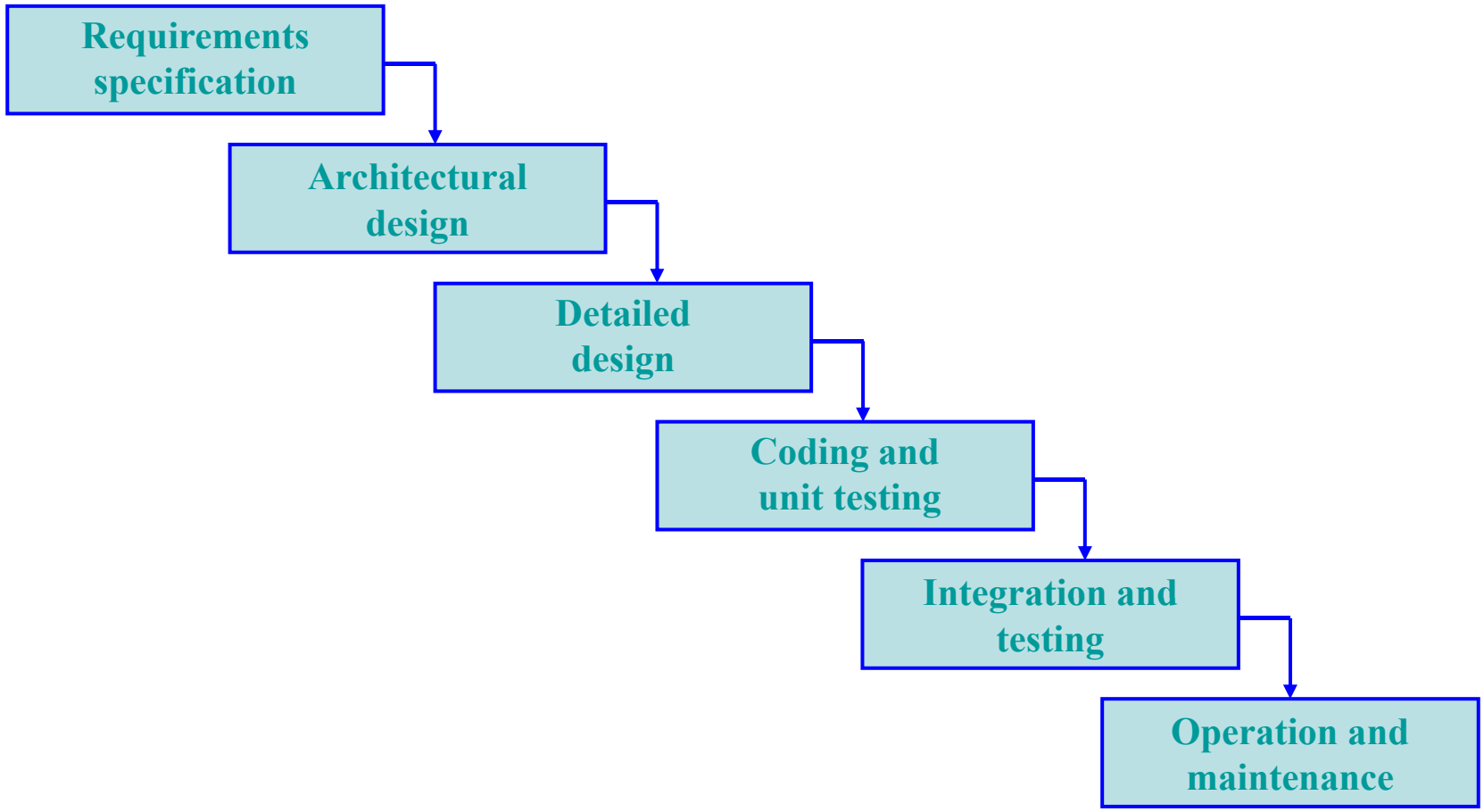
Relationship between ordinary quality and attractive quality



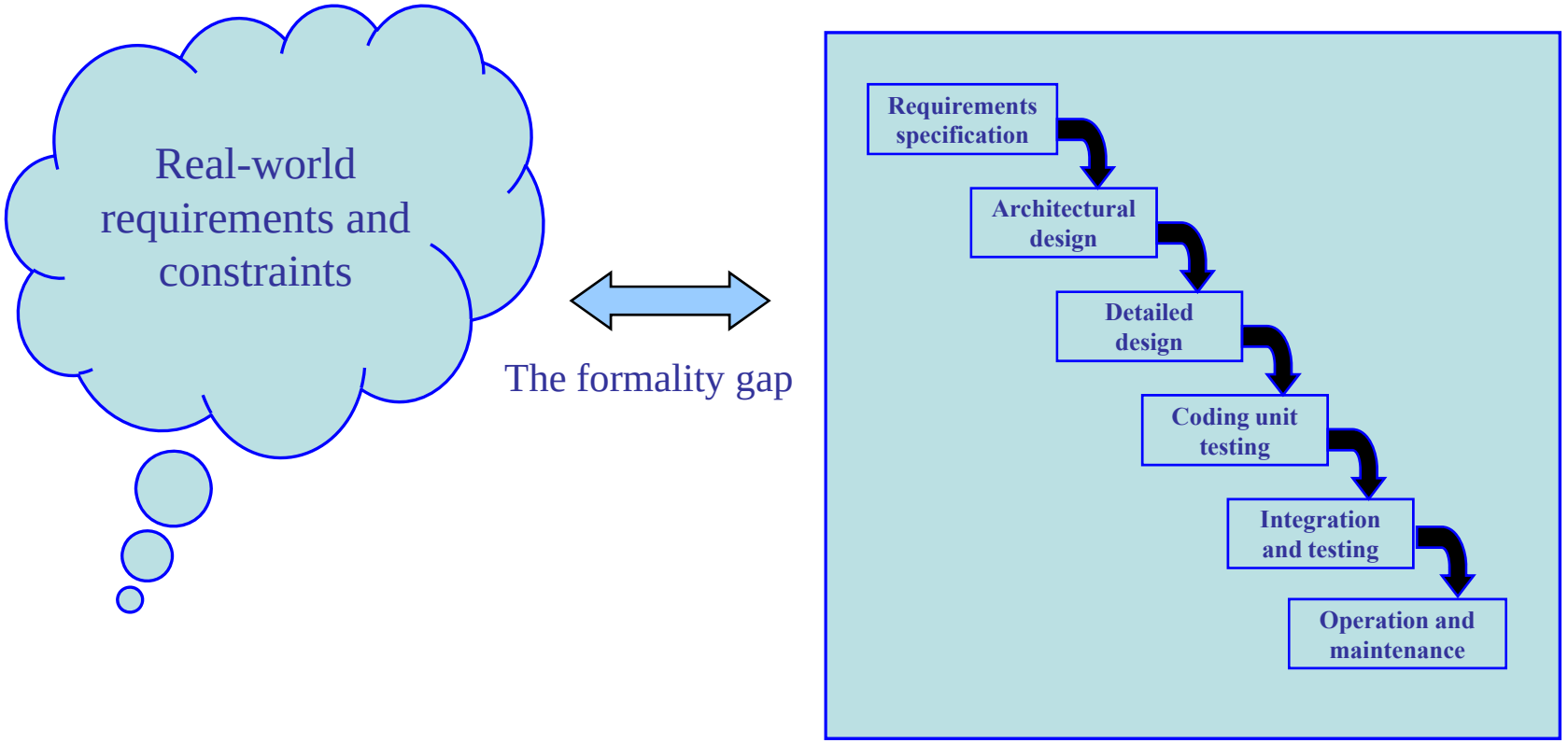
Comparison between usability and other factors

The HCI development methodology.

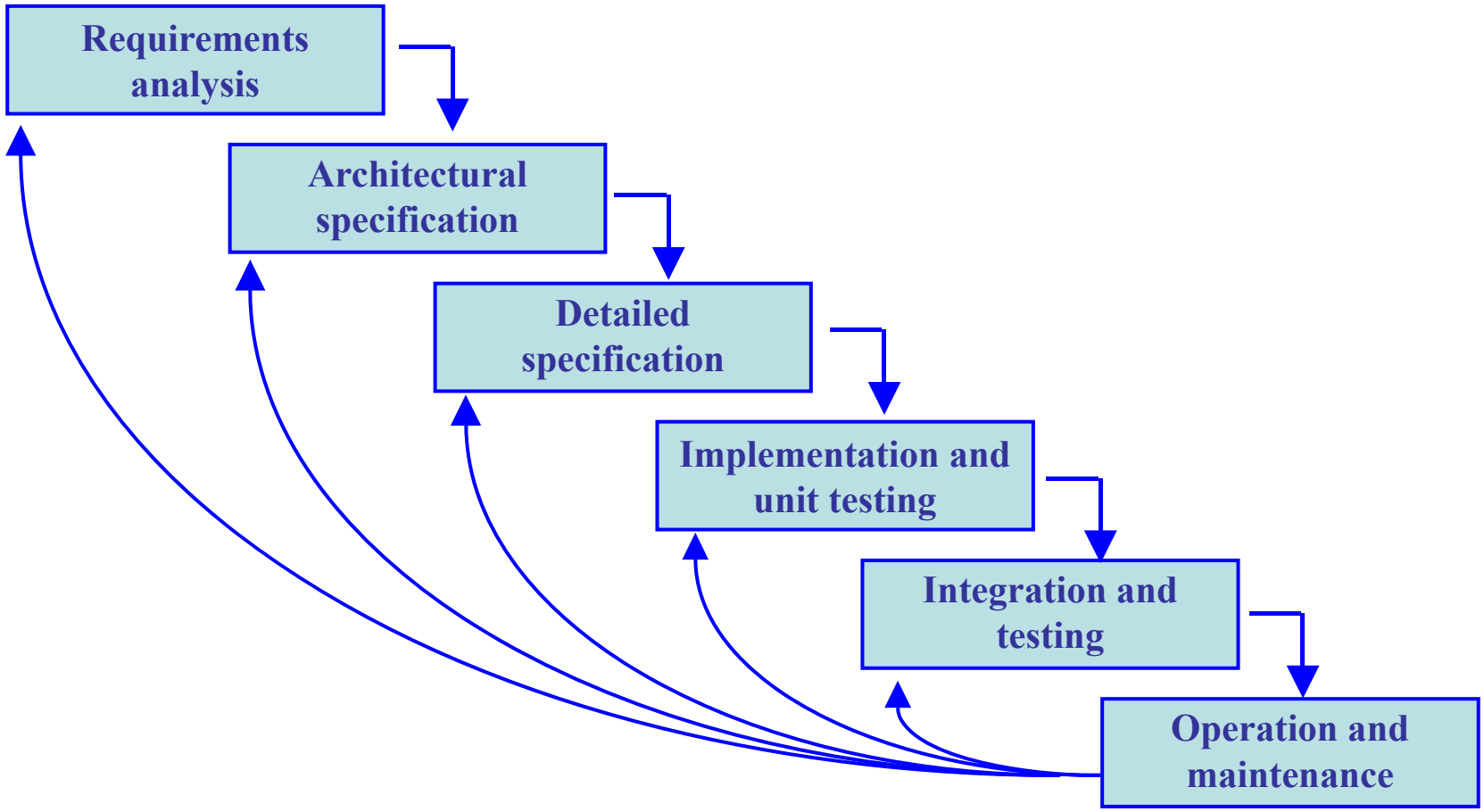




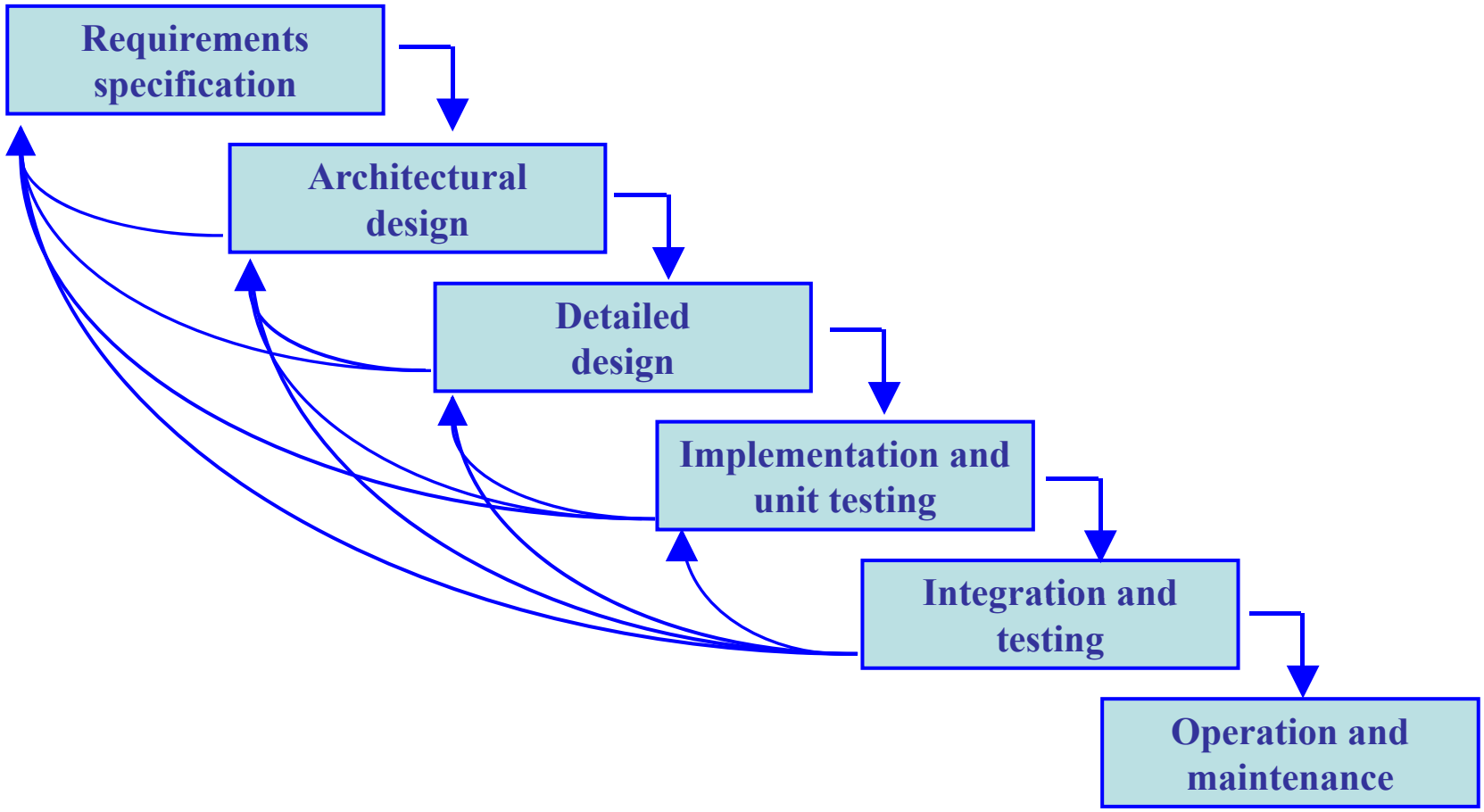
The activities in the waterfall model of the software life cycle



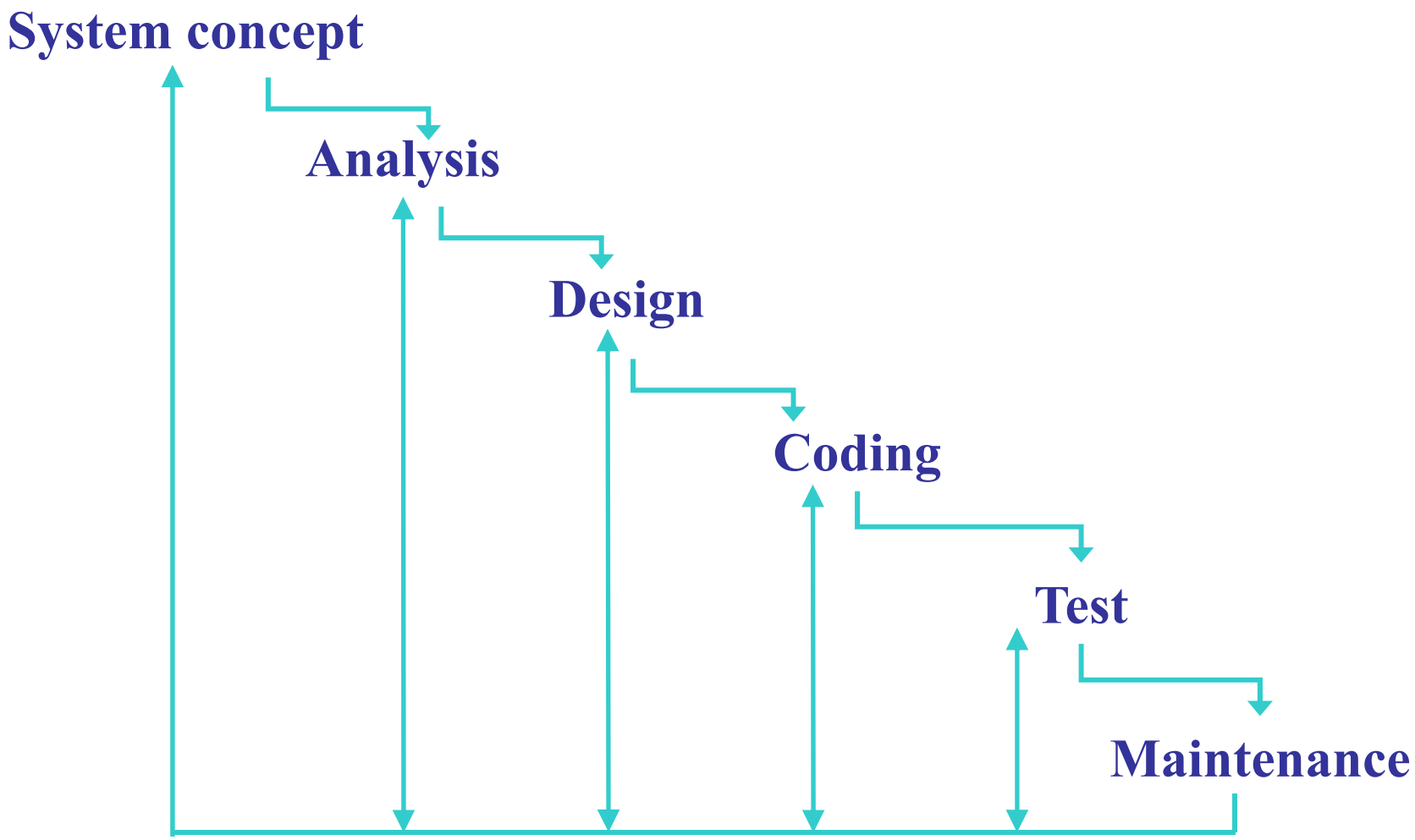
The formality gap between the real world and structured design



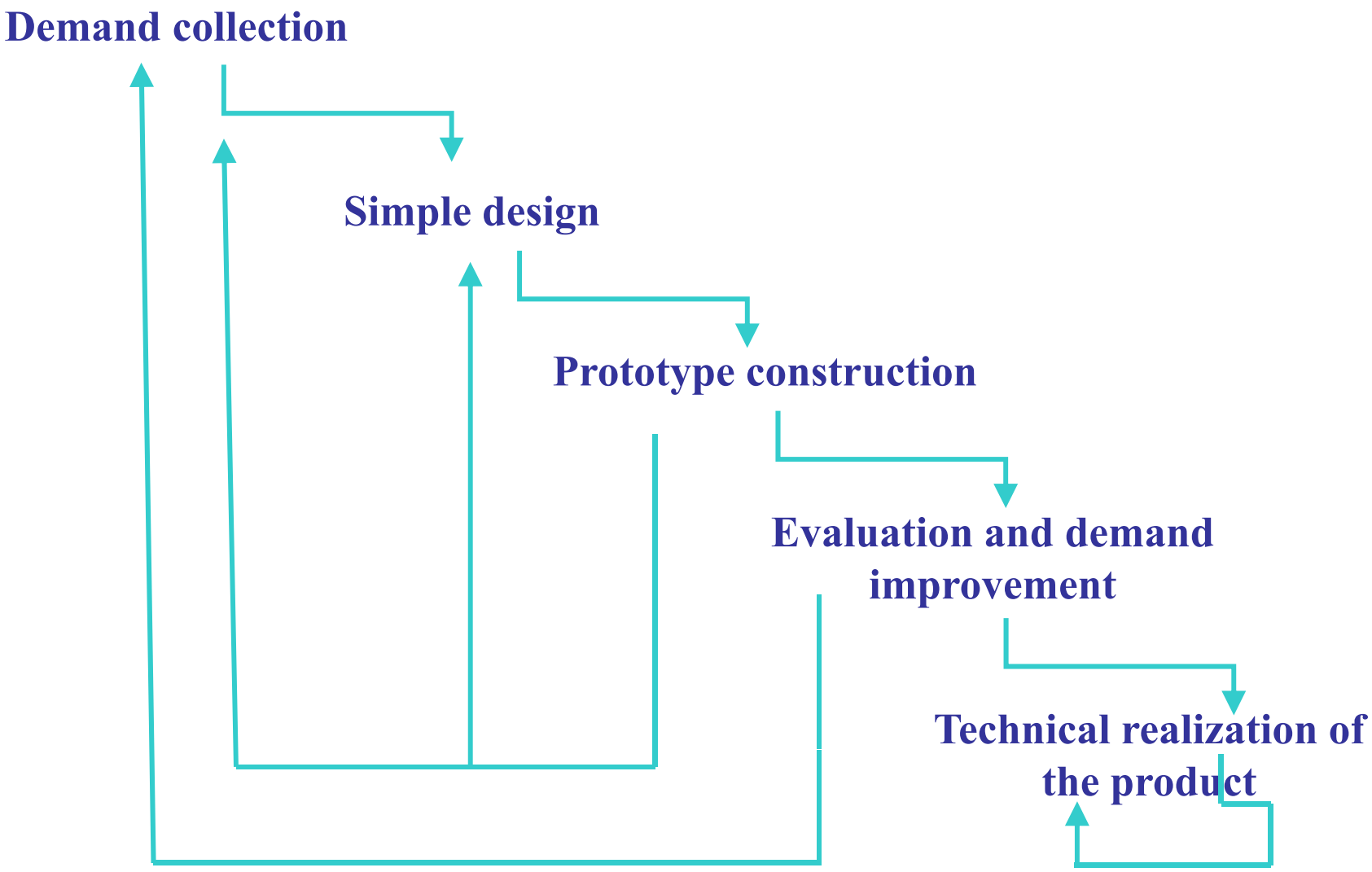
Feedback from maintenance activity to other design activities



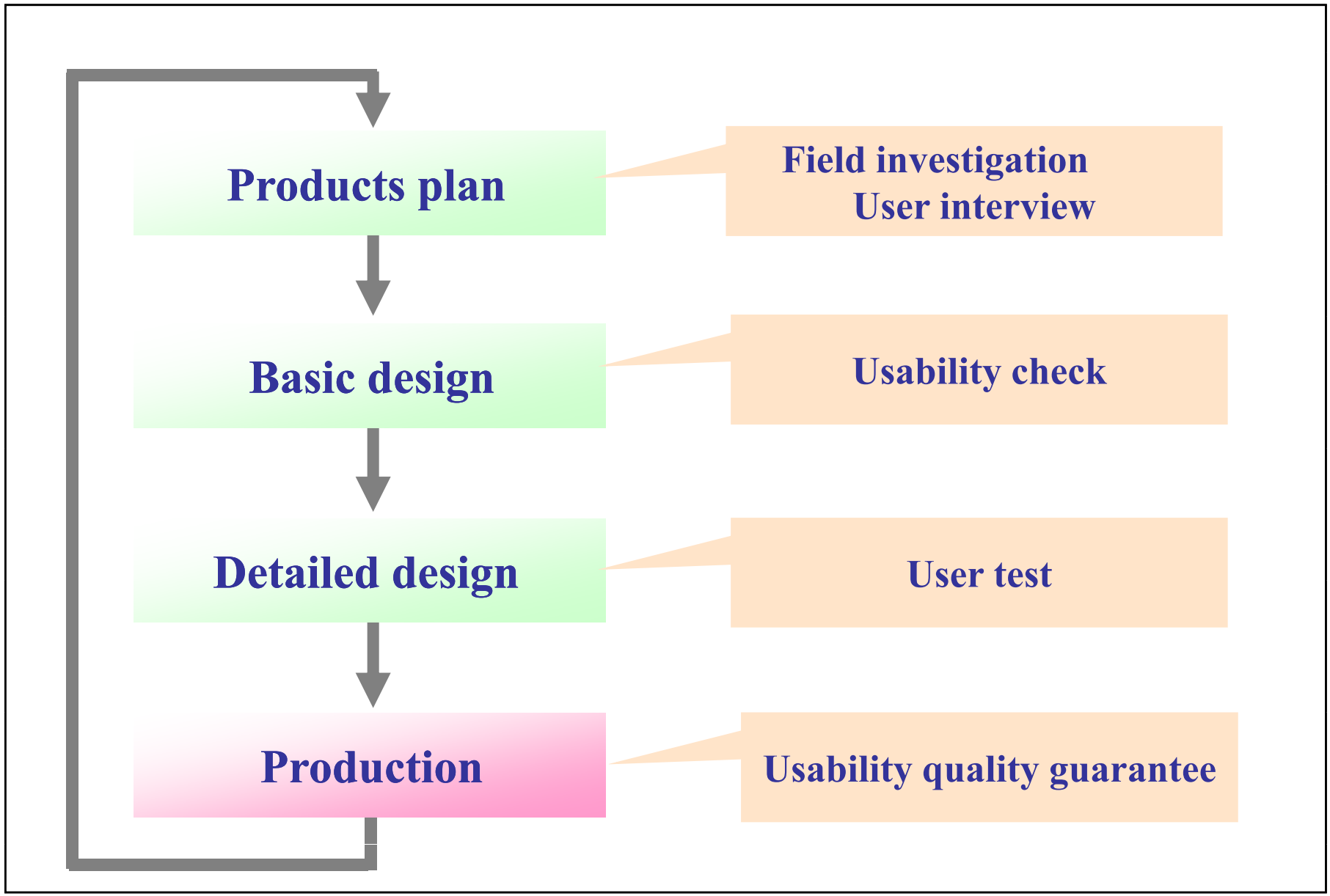
Representing iteration in the waterfall model



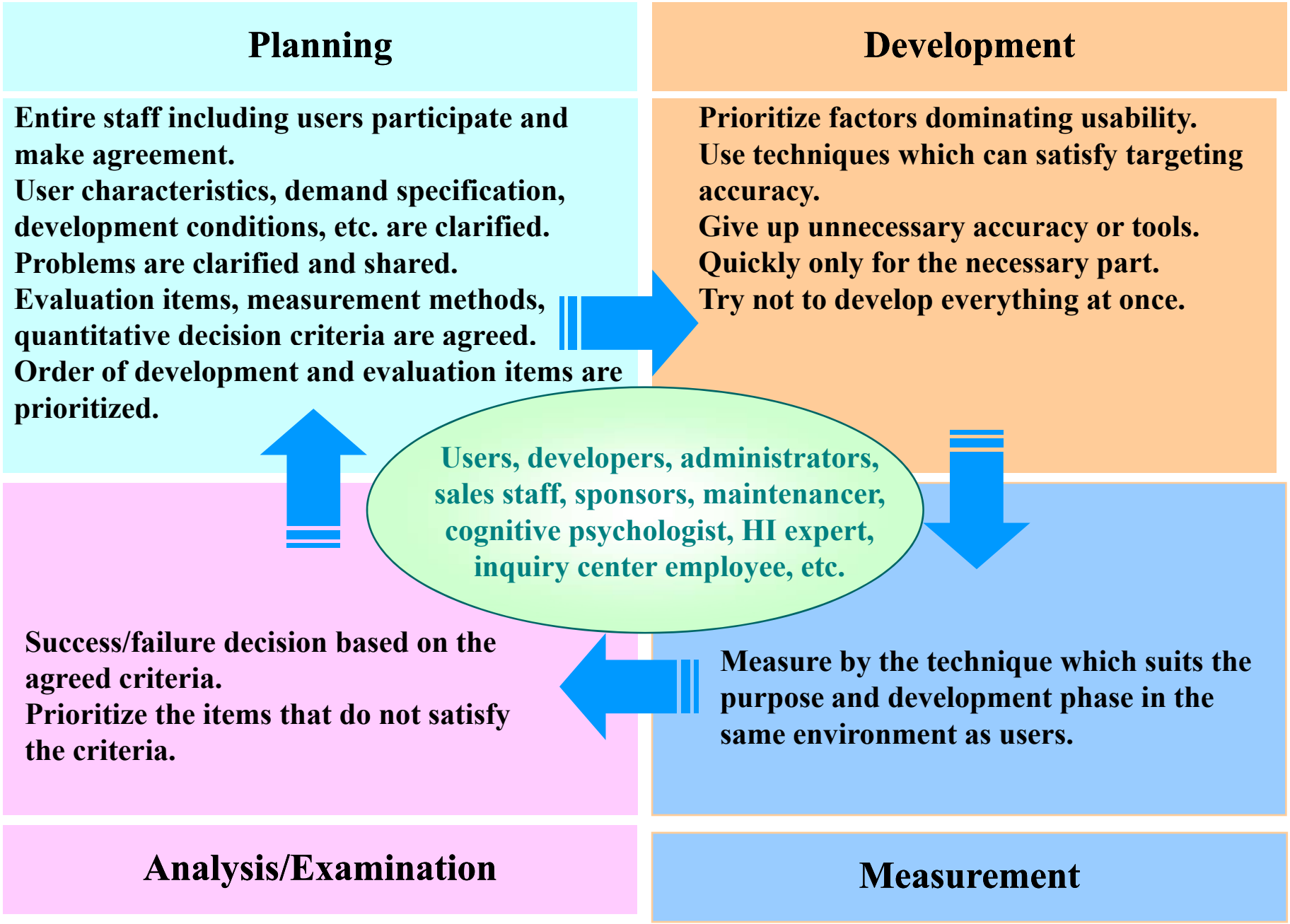
Waterfall model of system development



Prototyping model of design process



Product development and usability activities



Process of usability engineering

Design Principles and Guidelines

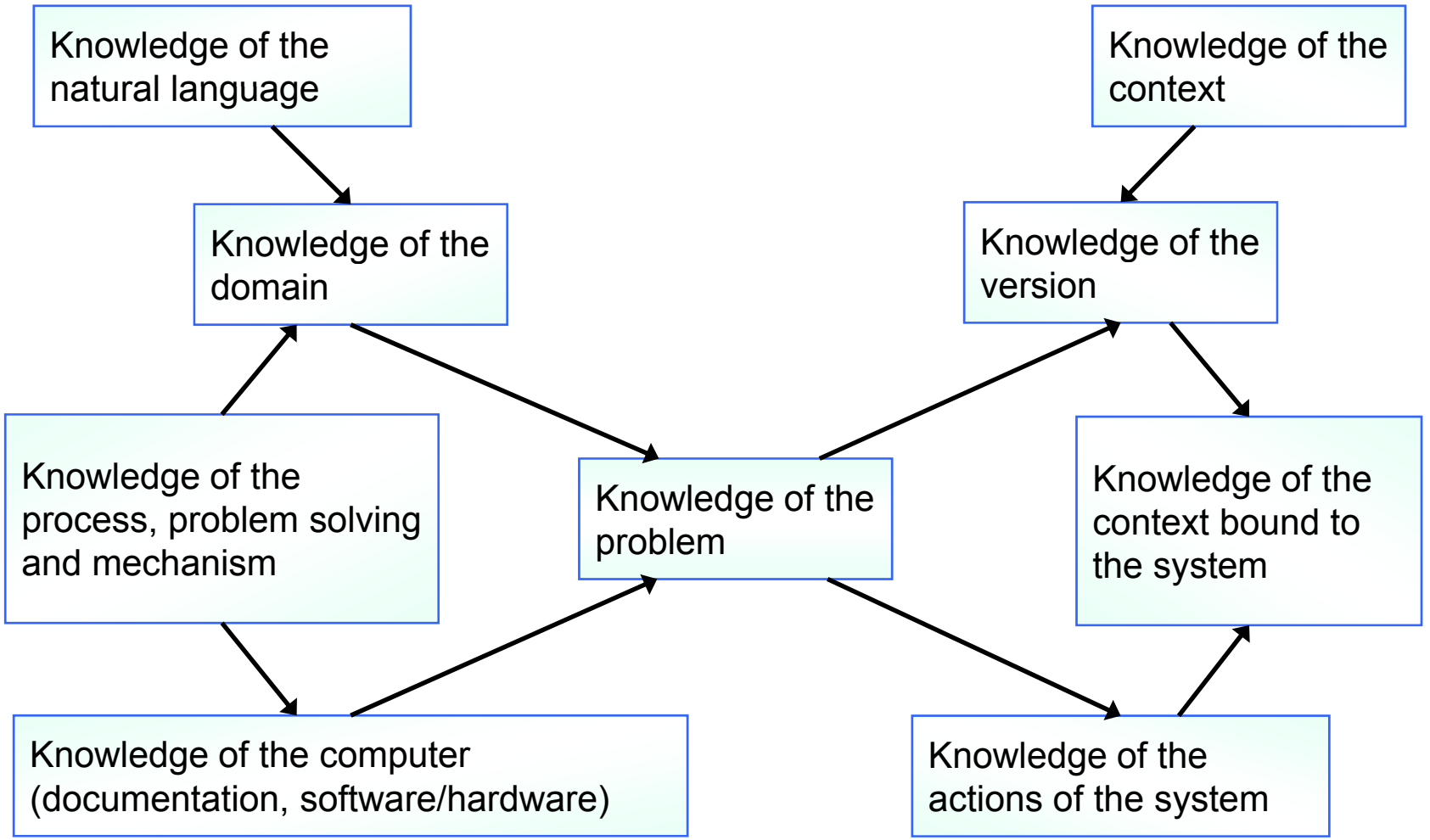
Design Principles

- Improve user's task performance and reduce their effort
- Strive for fit between the information representation needed and presented
- Direct and constrain user affordances to capture real-world knowledge
- Design for error
- Enable an enjoyable and satisfying interaction
- Promote trust
- Support diversity of users

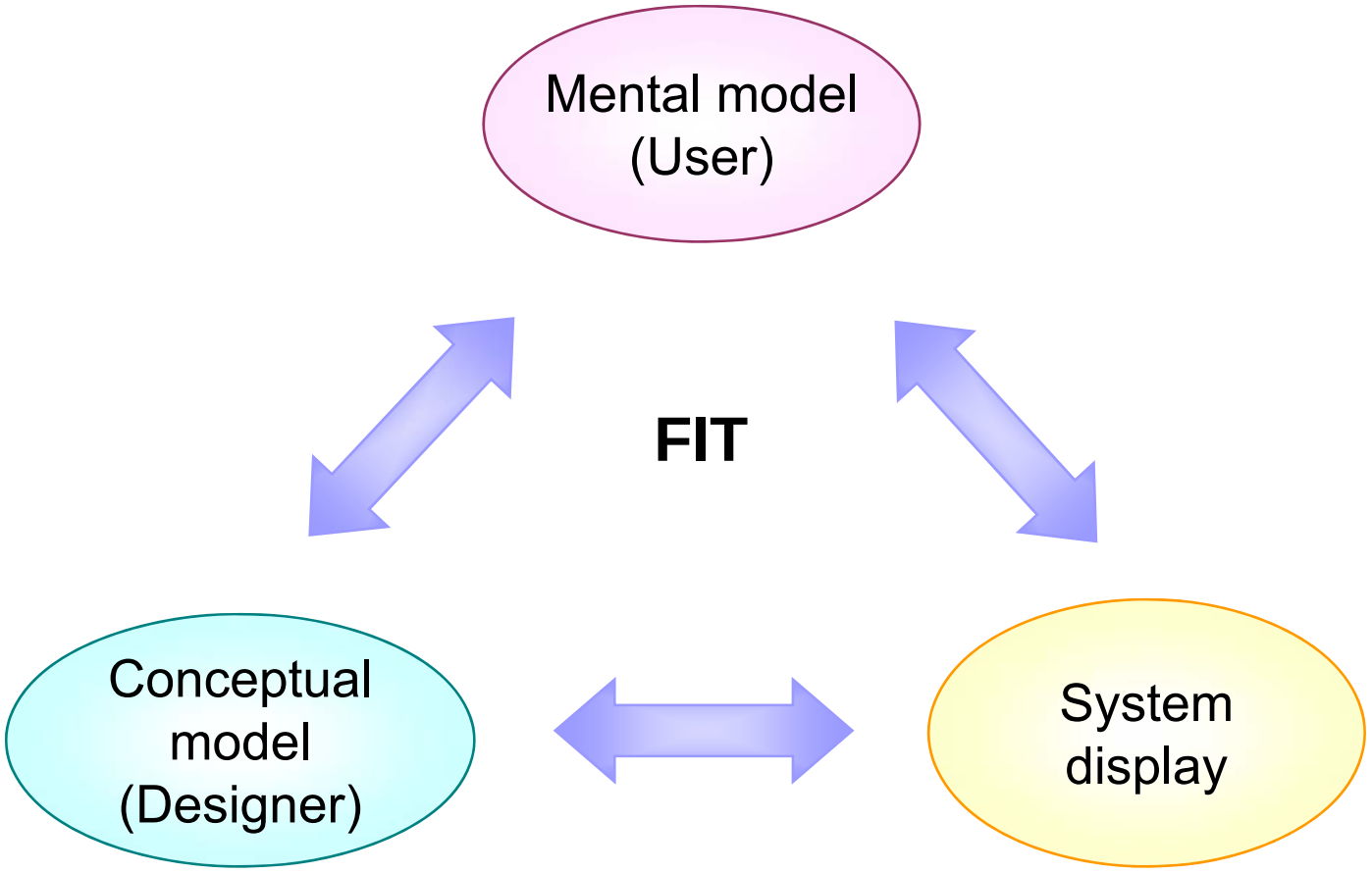
Design Guidelines

- Maintain consistent interaction
- Provide the user with control over the interaction, supported by feedback
- Use metaphors
- Use direct manipulation
- Design aesthetic interfaces

The Block interaction diagram (Hammond & Barnard)



Three models — the designer's conceptual model, the user's mental model, and the display of the system.

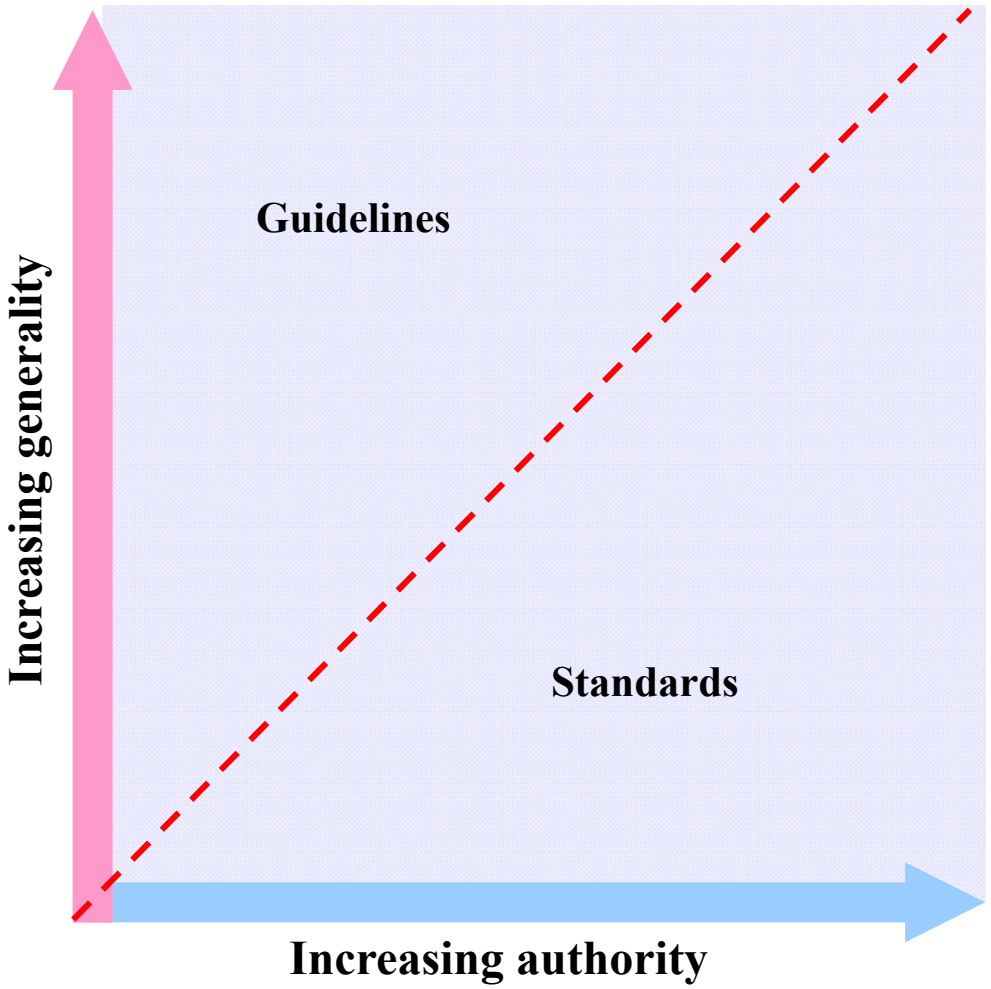


Interface design phases and general work flow

Design phase	Design work
Concept design	• Describing system concept and making development program • Grasp of supposed users, environment in use, and circumstances • Establishing function design and evaluation system • Description of scenario • Constructing metaphor and interface world
Realization design	• Layout interface technology and system • Design operation sequence • Design expression system（system design guide）
Detailed design	• Component design
Evaluation & revision	• Prototyping and evaluation • Revision of design guide, and design details • Maintenance and upgrade design

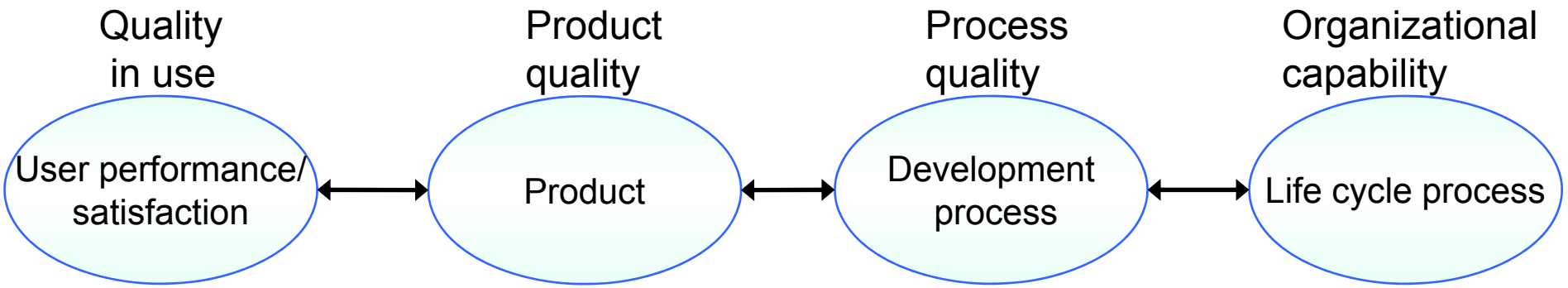
Effectiveness and problems of software design guidelines

	Effectiveness	Problems
For the person using the guideline	• Will have the skill to make HI design and gain confidence by understanding the content of the guideline. • The person who clings to particular design based on experiences and taste can seize the opportunity to gain flexibility in designing.	• Since useful software cannot necessarily be made by simply using the guidelines, it is not easy to have an incentive. • Because of the systematization of past knowledge, they are sometimes behind new technology, and not attractive. • Unless the users themselves update the guidelines, experiences and knowledge that designer obtained cannot be included.
For the jobs using the guideline	• Can check and improve the software efficiently in the middle of designing. • The guidelines can be used as development specifications of software. • The guidelines can be used as textbook to learn how to use the software.	• Of no use to plan a new software. • No consideration of practical design restriction, such as budget, personnel, and deadline. • Difficult to be translated to practical software design because of many general description.
For the environment using the guideline	• Since collected as a book, they are easy to carry. • Since massive knowledge is systematized, and relationship between the items are shown, necessary information can be easily found using database technology.	• Because of abstract explanation, users need to know the whole content to some degree to get appropriate guidelines. • Because of many similar items, it is difficult to find important ones.



Characterizing the authority and generality of standards and guidelines as design rules

Categories of HCI-related standards



ISO standard 9241

Usability:

The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

Effectiveness:

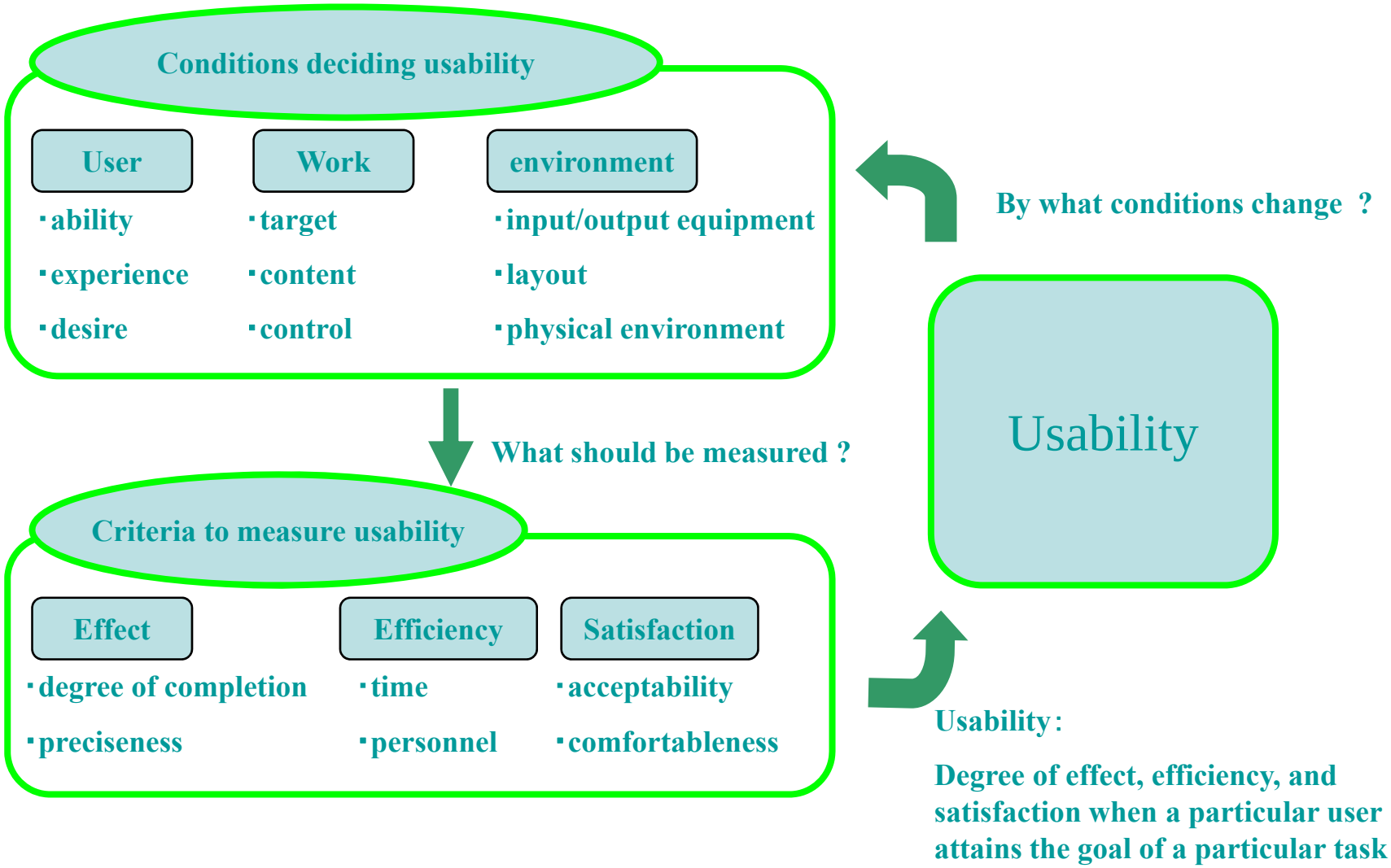
The accuracy and completeness with which specified users can achieve specified goals in particular environments.

Efficiency:

The resources expended in relation to the accuracy and completeness of goals achieved.

Satisfaction:

The comfort and acceptability of the work system to its users and other people affected by its use.

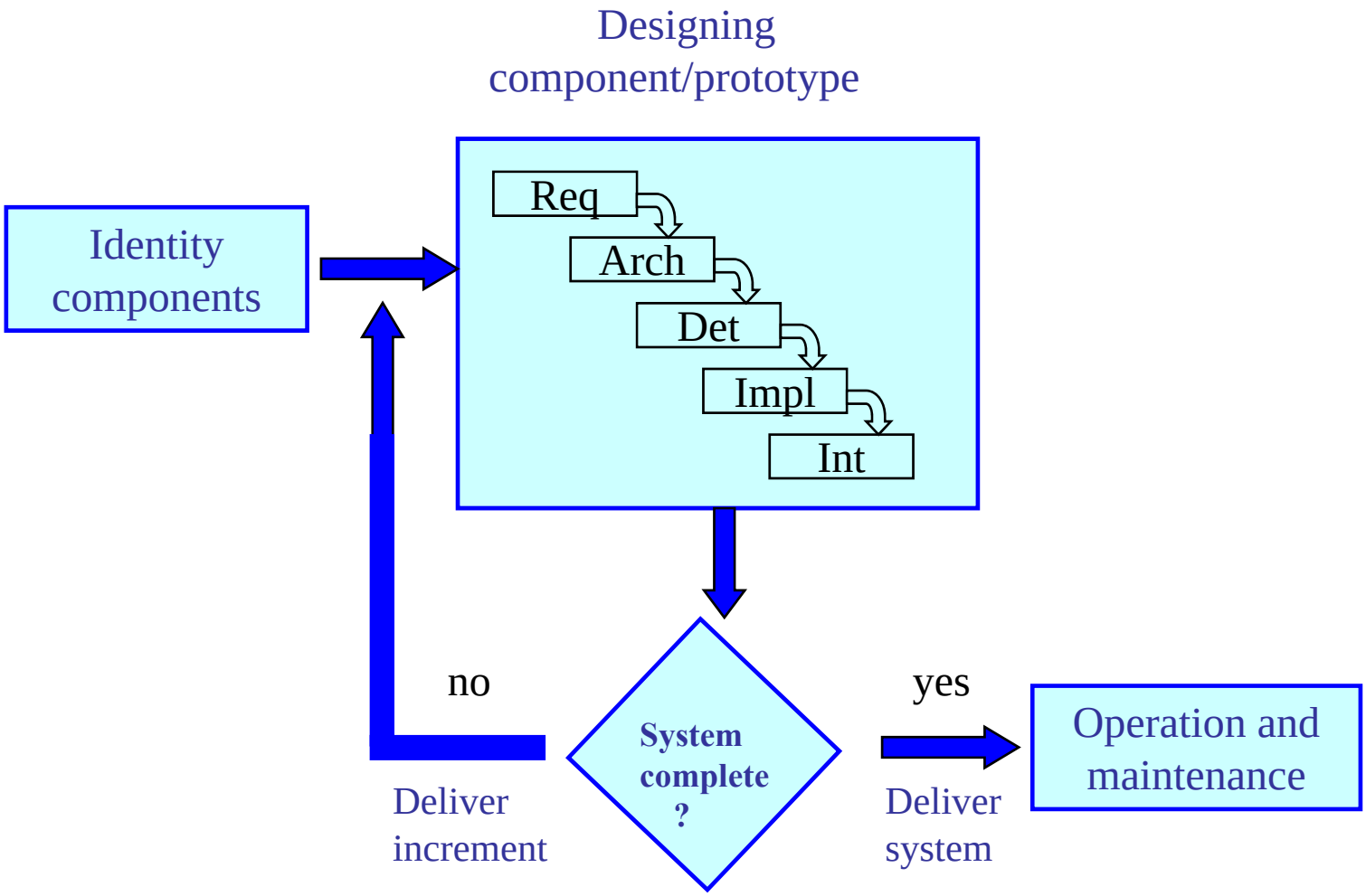


Conditions and criteria of usability

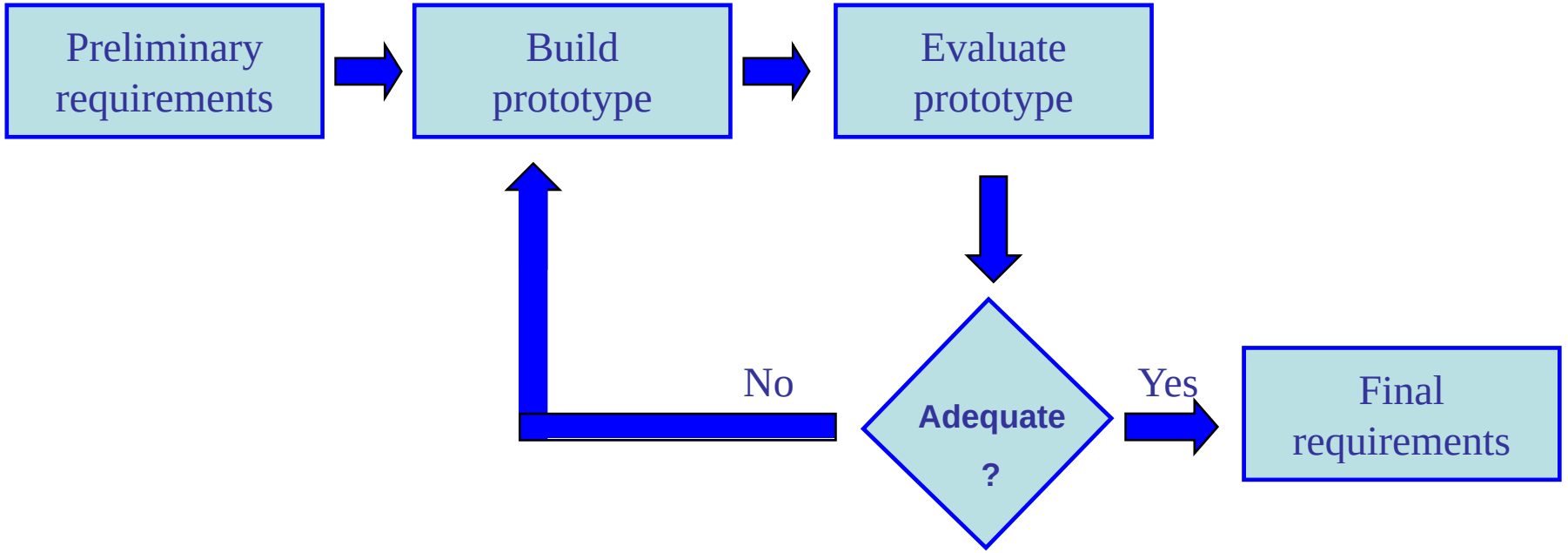
Criteria by which measuring methods can be determined (Whiteside, Bennett and Holtzblatt)

1. Time to complete a task
2. Percent of task completed
3. Percent of task completed per unit time
4. Ratio of successes to failures
5. Time spent in errors
6. Percent or number of errors
7. Percent or number of competitors better than it
8. Number of commands used
9. Frequency of help and documentation use
10. Percent of favorable / unfavorable user comments
11. Number of repetitions of failed commands
12. Number of runs of successes and failures
13. Number of times interface misleads the user
14. Number of good and bad features recalled by users
15. Number of available commands not invoked
16. Number of regressive behaviors
17. Number of users preferring your system
18. Number of times users need to work around a problem
19. Number of times the user is disputed from a work task
20. Number of times user loses control of the system
21. Number of times user expresses frustration or satisfaction

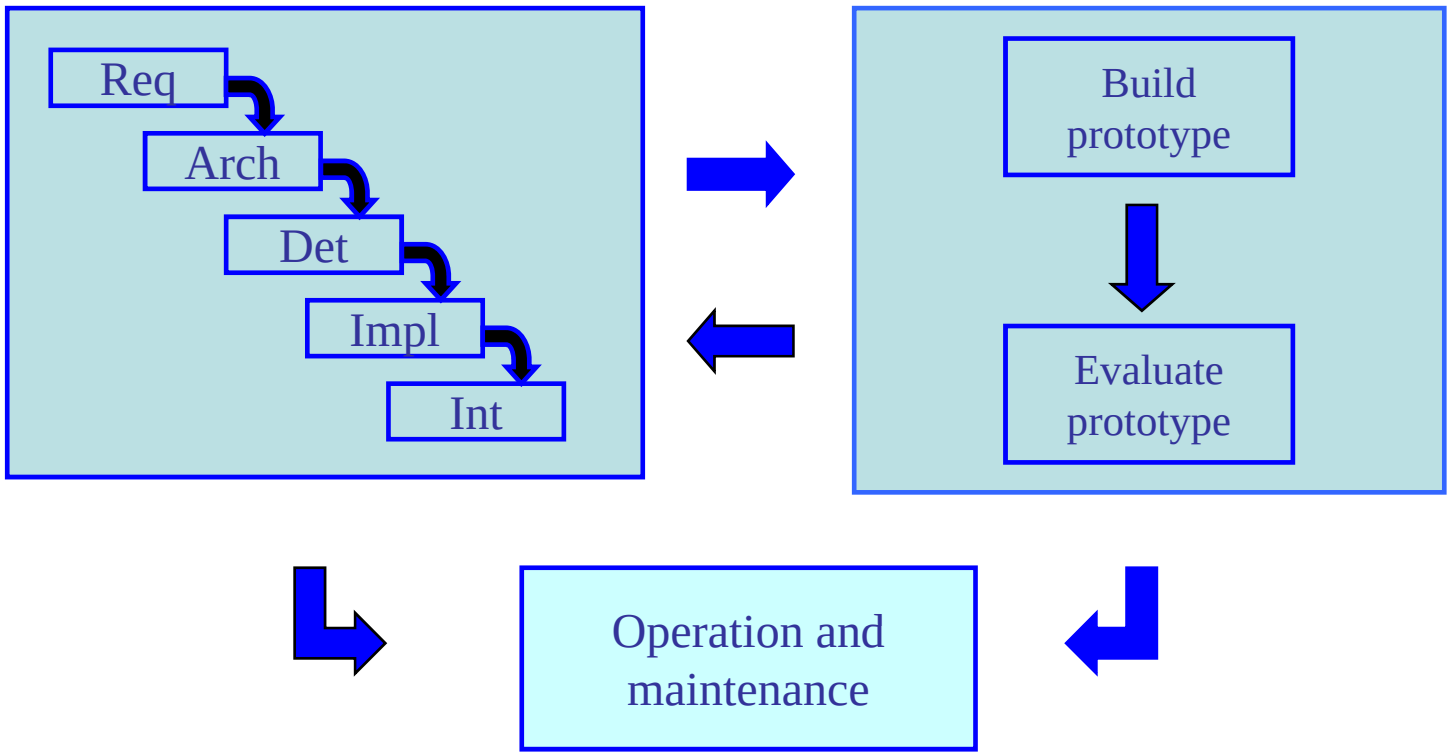
Examples of usability metrics from ISO 9241 (Committee Draft 9241-11.2)			
Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling



Incremental prototyping within the life cycle



Throw-away prototyping within requirements specification



Evolutionary prototyping throughout the life cycle

Adaptive multimodal interface

User model

- **Attribute（generation, work, taste, special features of machine use)**
- **User's knowledge**
- **Structure of user's goal/means**

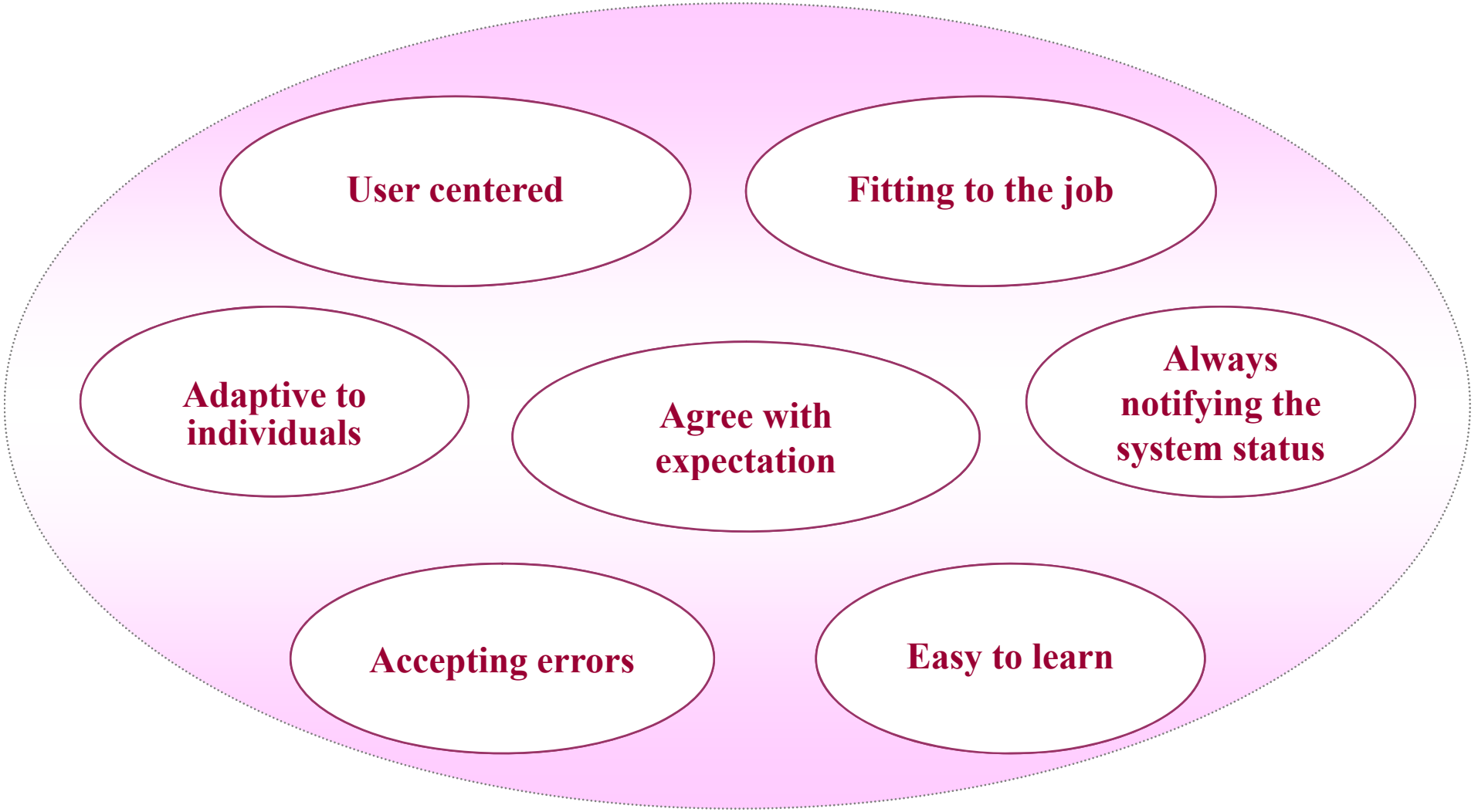
Discourse model

- **Grammar of multiple sentences, meaning structure**
- **Context**

Dialog model

- **Basic grammar of dialogue, semantic structure**

Basic principles of usable interface (dialogue system) design (ISO 1994)



Basic principles of usable interface (dialogue system) design (ISO1994)

1. Fitting to the job
   - Support the user to effectively and efficiently attain the task goals

2. Always notifying the system status
   - At every phase of dialogue, offer information as a feedback from system and user's demand so that the user can understand the system state

3. User centered
   - The user can change style and pace of the dialogue
   - The user feels that he/she is controlling the system
   - The user can select/adjust the input method and the speed of dialogue

4. Agree with expectation
   - The system is operated according to the user's requests which are based on knowledge, education, experiences about the task, and general custom.

5. Accepting errors

- Even if input errors occur, user can get expected results by minimum corrections.
- If every error is accepted, user can concentrate on the task without caring about errors.
- When errors occur, show the error and present a possible list of available inputs, so that the error can be easily corrected.

6. Adaptation to individuals

- Response can be changed according to the user's request and skill related to the task.
- Adaptive to the user's language, culture, knowledge, and experiences.

7. Easy to learn

- Support the user by offering help, tutorial, and guidance, so that he/she can easily learn the operations.

Grice's Cooperative Principle

"Make your contribution such as required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged" (Grice, 1975)

Grice's Cooperative Axioms

Conditions for the rational and efficient dialogues:

1. **Quantity**: Make your contribution as informative as is required. Do not make your contribution more informative than is required.
2. **Quality**: Do not say what you believe to be false. Do not say that for which you lack adequate evidence.
3. **Relation**: Be relevant. (Stay on topic.)
4. **Manner**: Avoid obscurity of expression. Avoid ambiguity. Be brief (avoid unnecessary prolixity). Be orderly.

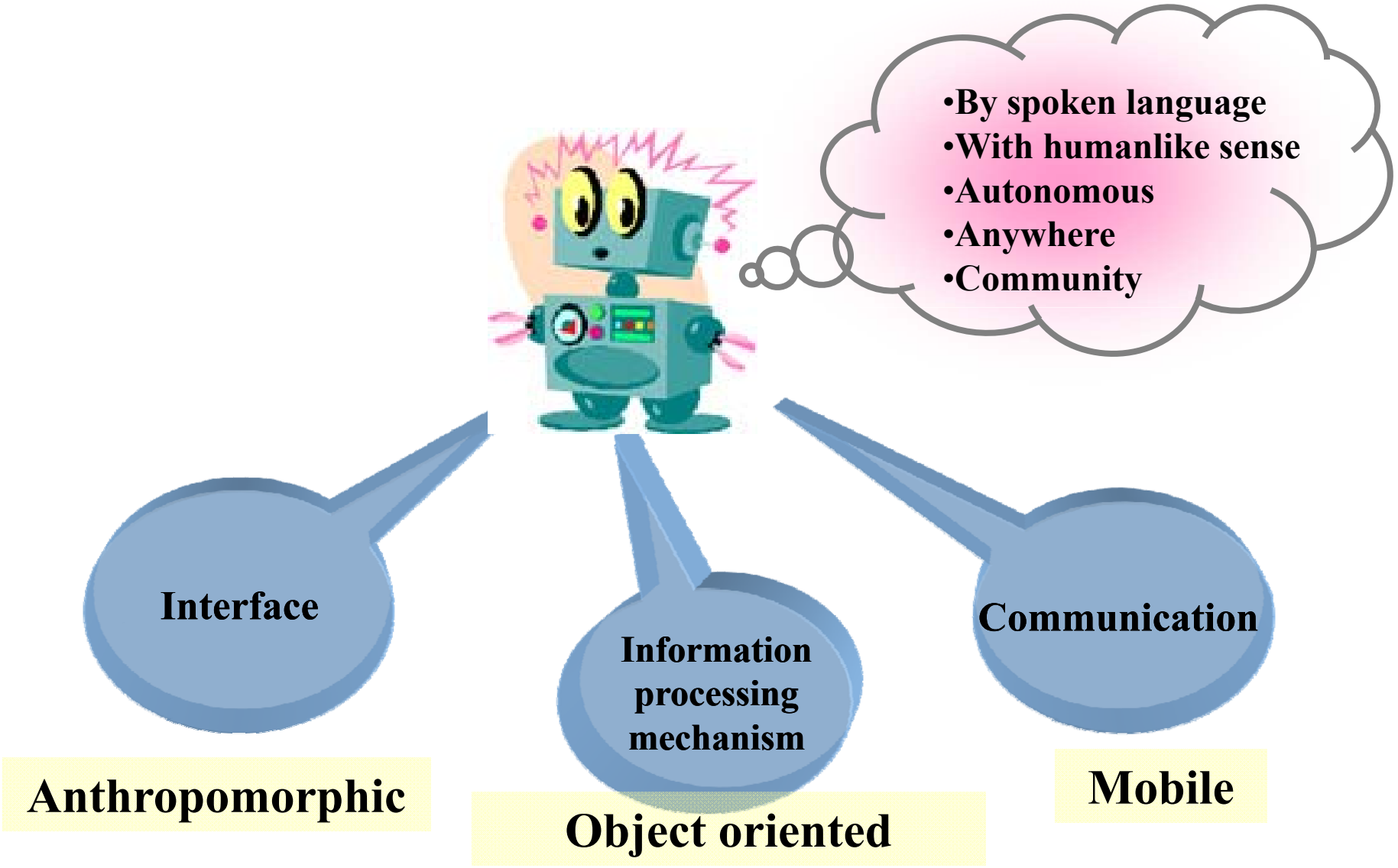
Evaluation items for speech dialogue systems (Nakagawa, 1998)	
1	Can natural utterances (ungrammatical sentences, various ways of speaking) be accepted ?
2	How accurate are the input sentences understood?
3	What is the grammatical and vocabulary coverage of the system ?
4	How is misunderstanding by the system solved ?
5	How many times of confirmation, re-input, paraphrasing are necessary ?
6	Are interval responses and barge-in accepted ?
7	Are vague sentences accepted ?
8	User-initiative, system-initiative, or mix-initiative?
9	Are system responses natural and easy to understand?
10	Is the quality of synthesized voice by the system good enough ?

Influential factors

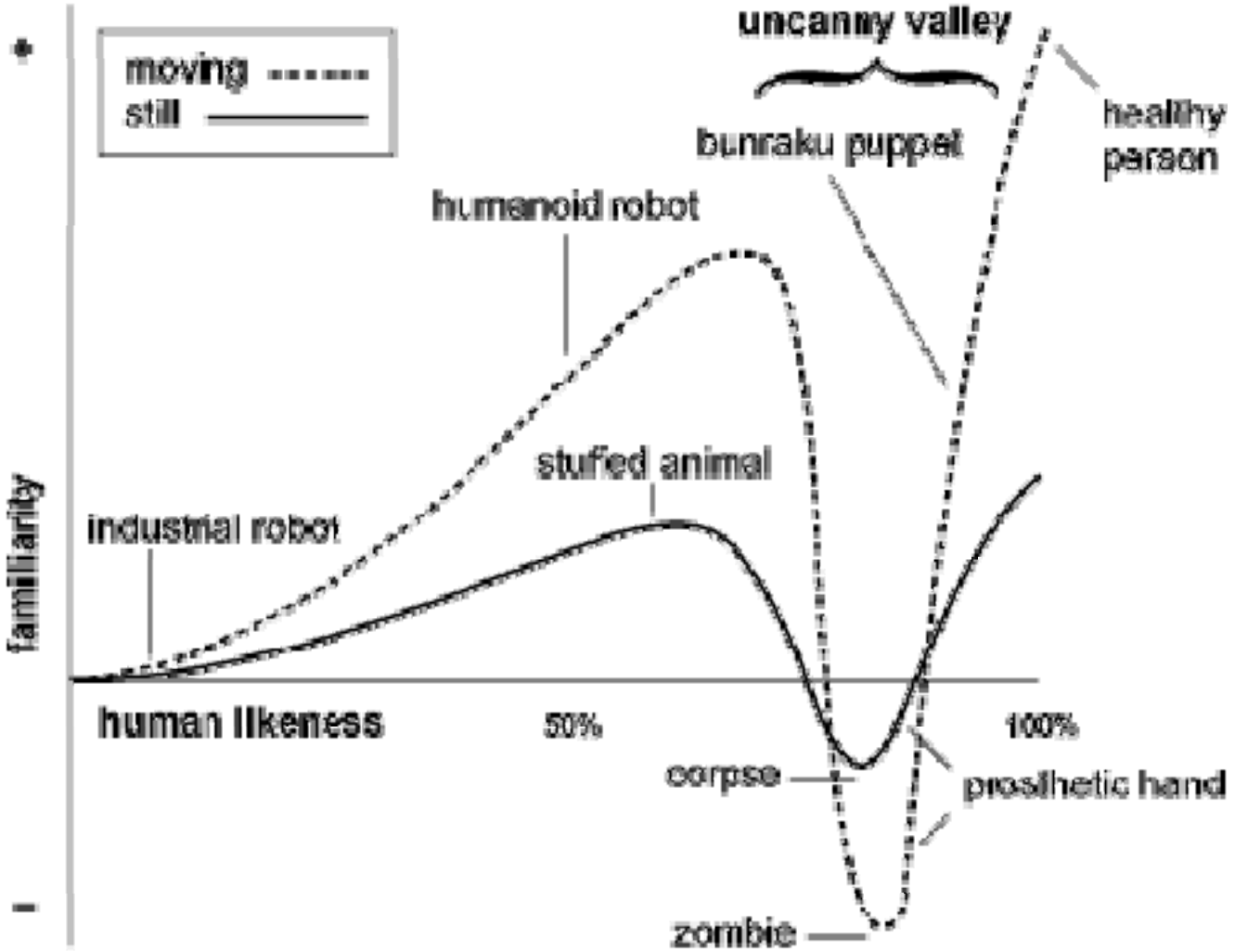
- User's familiarity with the system (a beginner or an expert ?)
- What input modalities are used in addition to speech? (keyboard, mouse, face image, etc.)
- What output modalities are used in addition to speech? (display, etc.)
- System response time

Anthropomorphic agent

- **Agent responding as if it were a human**
  - **Speech recognition, image recognition, speech synthesis**
- **Avatar**
  - **Action in a virtual space in behalf of a human**
  - **Virtual community**
- **Interactive art**



Framework of an agent system



Hypothesized emotional response of human subjects plotted against anthropomorphism of a robot